علم البيانات

عن طريق الامثلة

40 مشروع علم بيانات تم حلها وشرحها باستخدام بايثون

ترجمة واعداد: د. علاء طعيمة

بسمه تعالى

علم البيانات: عن طريق الامثلة

40 مشروع علم بيانات تم حلها وشرحها باستخدام بايثون

ترجمة واعداد:

د. علاء طعيمة

مقدمة المؤلف

تعد البيانات اليوم أداة ووقودًا للشركات لاكتساب رؤى مهمة وتحسين أدائها. سيطر علم البيانات على كل صناعة تقريبًافي العالم. لا توجد صناعةفي العالم اليوم لا تستخدم البيانات.

بصفتك مبتدئًافي علم البيانات، من الصعب فهم جميع المفاهيم التي تتعلمها دون تنفيذهافي مجموعة بيانات. سيساعدك العملفي مشاريع علم البيانات ودراسات الحالة على تحسين مهاراتكفي علم البيانات. إذا كنت تكافح من أجل ابتكار أفكار لمشروع علم البيانات وكيفية بدء مشروع علم البيانات وإنهائه، فهذا الكتاب مناسب لك.في هذه الكتاب، سوف يأخذك المؤلف من خلال قائمة مشاريع علم البيانات باستخدام بايثون والتي ستساعدك على تعلم وتنفيذ جميع مفاهيم علم البيانات باستخدام لغة برمجة بايثون.

لقد حاولت قدر المستطاع ان اترجم المشاريع الأكثر طرحاًفي مجال علم البيانات مع الشرح المناسب والكافي، ومع هذا يبقى عملاً بشرياً يحتمل النقص، فاذاكان لديك أي ملاحظات حول هذا الكتاب، فلا تتردد بمراسلتنا عبر بريدنا الالكتروني alaa.taima@qu.edu.iq .

نأمل ان يساعد هذا الكتاب كل من يريد ان يدخلفي مجالات علم البيانات ومساعدة القارئ العربي على تعلم هذا المجال. اسأل الله التوفيقفي هذا العمل لأثراء المحتوى العربي الذي يفتقر أشد الافتقار إلى محتوى جيد ورصينفي مجال علم البيانات. ونرجو لك الاستمتاع مع الكتاب ولا تنسونا من صالح الدعاء.

د. علاء طعيمة

كلية علوم الحاسوب وتكنولوجيا المعلومات

جامعة القادسية

العراق

المحتويات

1) التنبؤ بالطقس باستخدام بايثون Weather Forecasting using Python

في علم البيانات (Data Science)، التنبؤ بالطقس هو تطبيق للتنبؤ بالسلاسل الزمنية (Time Series Forecasting) حيث نستخدم بيانات وخوارزميات السلاسل الزمنية لعمل تنبؤات لوقت معين. إذا كنت تريد معرفة كيفية التنبؤ بالطقس باستخدام مهاراتكفي علم البيانات، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالطقس (weather forecasting) باستخدام بايثون.

التنبؤ بالطقس

التنبؤ بالطقس هو مهمة التنبؤ بأحوال الطقس لموقع ووقت معين. باستخدام بيانات وخوارزميات الطقس، من الممكن التنبؤ بأحوال الطقس لعدد n من الأيام القادمة.

للتنبؤ بالطقس باستخدام بايثون، نحتاج إلى مجموعة بيانات تحتوي على بيانات الطقس التاريخية بناءً على موقع معين. لقد وجدت مجموعة بيانات على Kaggle استنادًا إلى بيانات الطقس اليوميةفي نيودلهي. يمكننا استخدام مجموعة البيانات هذه لمهمة التنبؤ بالطقس. يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، ستتعلم كيف يمكننا تحليل الطقس والتنبؤ به باستخدام بايثون.

تحليل بيانات الطقس باستخدام بايثون

الآن لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات التي نحتاجها:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

data = pd.read_csv("DailyDelhiClimateTrain.csv")
print(data.head())
```

	date	meantemp	humidity	wind_speed	meanpressure
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000

دعونا نلقي نظرة على الإحصائيات الوصفية لهذه البيانات قبل المضي قدمًا:

```
print(data.describe())
```

```
       meantemp     humidity    wind_speed  meanpressure
count  1462.000000  1462.000000  1462.000000  1462.000000
mean     25.495521    60.771702     6.802209  1011.104548
std       7.348103    16.769652     4.561602   180.231668
min       6.000000    13.428571     0.000000    -3.041667
25%      18.857143    50.375000     3.475000  1001.580357
50%      27.714286    62.625000     6.221667  1008.563492
75%      31.305804    72.218750     9.238235  1014.944901
max      38.714286   100.000000    42.220000  7679.333333
```

دعنا الآن نلقي نظرة على المعلومات المتعلقة بجميع الأعمدةفي مجموعة البيانات:

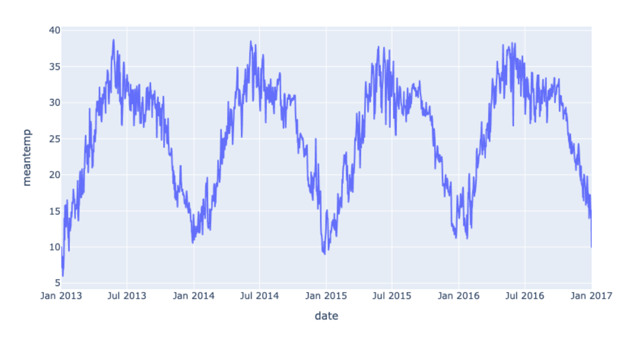
```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1462 entries, 0 to 1461
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   date          1462 non-null   object
 1   meantemp      1462 non-null   float64
 2   humidity      1462 non-null   float64
 3   wind_speed    1462 non-null   float64
 4   meanpressure  1462 non-null   float64
dtypes: float64(4), object(1)
memory usage: 57.2+ KB
```

لا يحتوي عمود التاريخ (date column) في مجموعة البيانات هذه على نوع بيانات التاريخ والوقت. سنقوم بتغييره عند الحاجة. دعونا نلقي نظرة على متوسط درجة الحرارة (mean temperature)في دلهي على مر السنين:

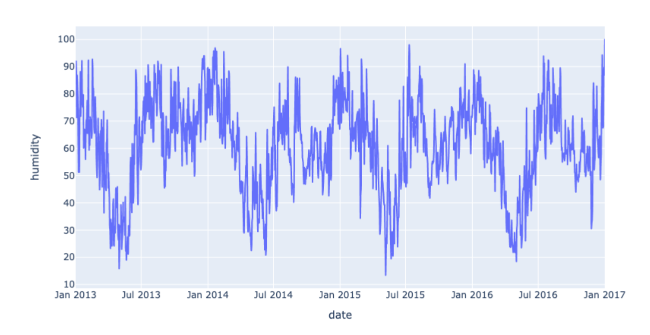
```
figure = px.line(data, x="date",
                 y="meantemp",
                 title='Mean Temperature in Delhi Over the
Years')
figure.show()
```





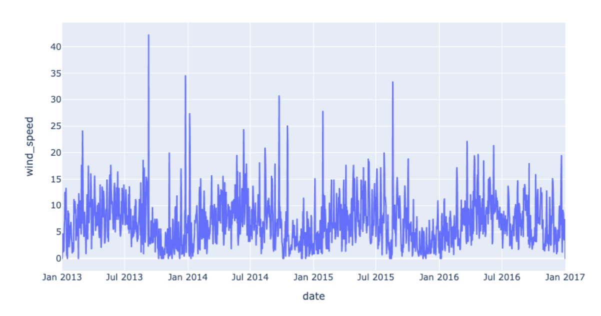
الآن دعونا نلقي نظرة على الرطوبة (humidity)في دلهي على مر السنين:

```
figure = px.line(data, x="date ,"
                 y="humidity ,"
                 title='Humidity in Delhi Over the Years('
figure.show()
```



الآن دعونا نلقي نظرة على سرعة الرياح (wind speed)في دلهي على مر السنين:

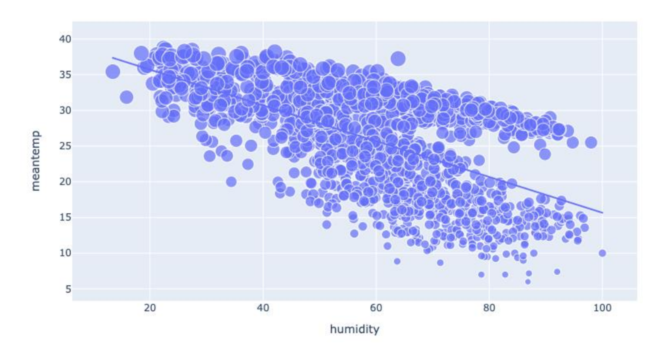
```
figure = px.line(data, x="date ,"
                 y="wind_speed ,"
                 title='Wind Speed in Delhi Over the Years('
figure.show()
```



حتى عام 2015، كانت سرعة الرياح أعلى خلال الرياح الموسمية (أغسطس وسبتمبر) وتراجع الرياح الموسمية (ديسمبر ويناير). بعد عام 2015، لم تكن هناك حالات شاذةفي سرعة الرياح خلال الرياح الموسمية. دعنا الآن نلقي نظرة على العلاقة بين درجة الحرارة والرطوبة:

```
figure = px.scatter(data_frame = data, x="humidity,"
                    y="meantemp", size="meantemp ,"
                    trendline="ols ,"
                    title = "Relationship Between Temperature
and Humidity("
figure.show()
```





هناك علاقة سلبية بين درجة الحرارة والرطوبةفي دلهي. وهذا يعني أن ارتفاع درجة الحرارة يؤدي إلى انخفاض الرطوبة وانخفاض درجة الحرارة يؤدي إلى ارتفاع نسبة الرطوبة.

تحليل تغير درجة الحرارة

الآن دعونا نحلل التغيرفي درجة الحرارةفي دلهي على مر السنين. بالنسبة لهذه المهمة، سأقوم أولاً بتحويل نوع بيانات عمود التاريخ إلى التاريخ والوقت. ثم سأضيف عمودين جديدينفي مجموعة البيانات لقيم السنة والشهر.

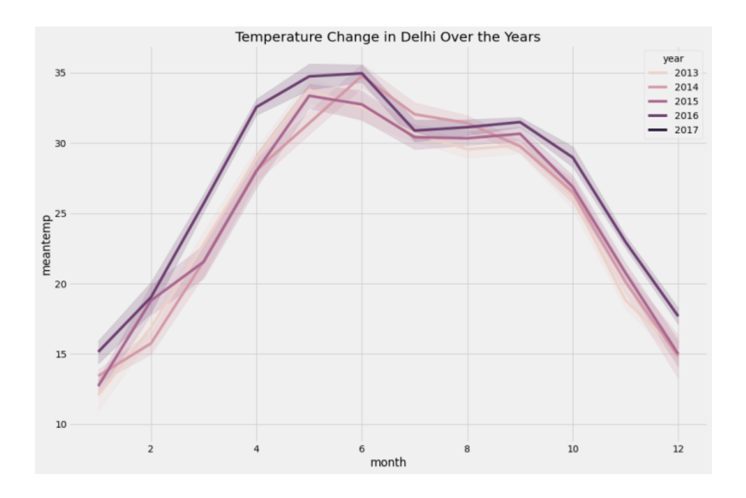
إليك كيفية تغيير نوع البيانات واستخراج بيانات السنة والشهر من عمود التاريخ:

```
data["date"] = pd.to_datetime(data["date"], format = '%Y-%m-
%d')
data['year'] = data['date'].dt.year
data["month"] = data["date"].dt.month
print(data.head())
```

	date	meantemp	humidity	wind_speed	meanpressure	year	month
0	2013-01-01	10.000000	84.500000	0.000000	1015.666667	2013	1
1	2013-01-02	7.400000	92.000000	2.980000	1017.800000	2013	1
2	2013-01-03	7.166667	87.000000	4.633333	1018.666667	2013	1
3	2013-01-04	8.666667	71.333333	1.233333	1017.166667	2013	1
4	2013-01-05	6.000000	86.833333	3.700000	1016.500000	2013	1

الآن دعونا نلقي نظرة على تغير درجة الحرارةفي دلهي على مر السنين:

```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(15, 10))
plt.title("Temperature Change in Delhi Over the Years")
sns.lineplot(data = data, x='month', y='meantemp', hue='year')
plt.show()
```



على الرغم من أن عام 2017 لم يكن الأكثر سخونةفي الصيف، يمكننا أن نرى ارتفاعًافي متوسط درجة حرارة دلهي كل عام.

التنبؤ بالطقس باستخدام بايثون

الآن دعنا ننتقل إلى مهمة التنبؤ بالطقس. سأستخدم نموذج (Facebook prophet) لهذه المهمة. يعد نموذج Facebook prophet أحد أفضل التقنيات للتنبؤ بالسلاسل الزمنية. إذا لم تستخدم هذا النموذج من قبل، فيمكنك تثبيته على نظامك باستخدام الأمر المذكور أدناهفي موجه الأوامر أو التيرمينال:

```
pip install prophet
```

يقبل نموذج prophet بيانات الوقت المسماة "ds" والتسميات على أنها "y". فلنحول البيانات إلى هذا التنسيق:

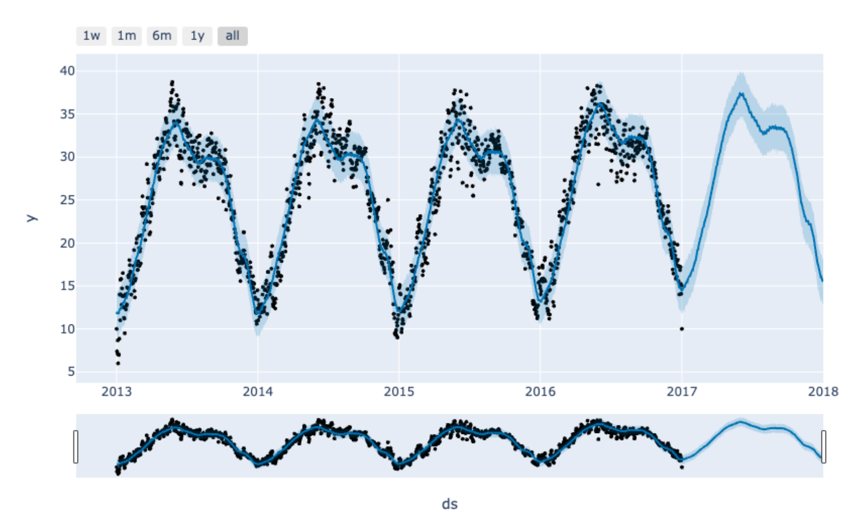
```
forecast_data = data.rename(columns = {"date": "ds ",
"                                     meantemp": "y"})
print(forecast_data)
```

```
            ds          y   humidity  wind_speed  meanpressure  year  month
0   2013-01-01  10.000000  84.500000    0.000000   1015.666667  2013      1
1   2013-01-02   7.400000  92.000000    2.980000   1017.800000  2013      1
2   2013-01-03   7.166667  87.000000    4.633333   1018.666667  2013      1
3   2013-01-04   8.666667  71.333333    1.233333   1017.166667  2013      1
4   2013-01-05   6.000000  86.833333    3.700000   1016.500000  2013      1
...        ...        ...        ...         ...           ...   ...    ...
1457 2016-12-28 17.217391  68.043478    3.547826   1015.565217  2016     12
1458 2016-12-29 15.238095  87.857143    6.000000   1016.904762  2016     12
1459 2016-12-30 14.095238  89.666667    6.266667   1017.904762  2016     12
1460 2016-12-31 15.052632  87.000000    7.325000   1016.100000  2016     12
1461 2017-01-01 10.000000 100.000000    0.000000   1016.000000  2017      1

[1462 rows x 7 columns]
```

الآن فيما يلي كيفية استخدام نموذج Facebook prophet للتنبؤ بالطقس باستخدام بايثون:

```
from prophet import Prophet
from prophet.plot import plot_plotly, plot_components_plotly
model = Prophet()
model.fit(forecast_data)
forecasts = model.make_future_dataframe(periods=365)
predictions = model.predict(forecasts)
plot_plotly(model, predictions)
```



هذه هي الطريقة التي يمكنك من خلالها تحليل الطقس والتنبؤ به باستخدام بايثون.

الملخص

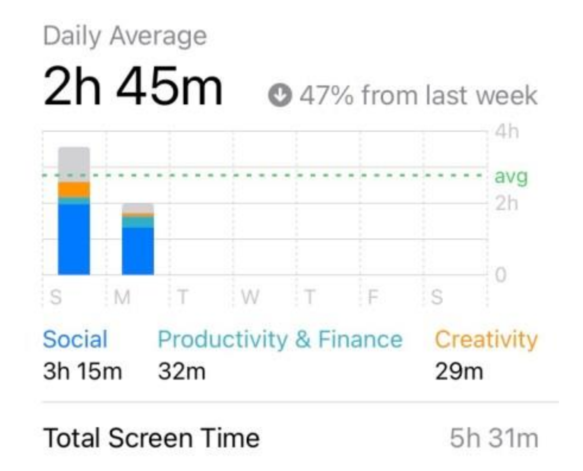
التنبؤ بالطقس هو مهمة التنبؤ بأحوال الطقس لموقع ووقت معين. باستخدام بيانات وخوارزميات الطقس، من الممكن التنبؤ بأحوال الطقس لعدد n من الأيام القادمة. أتمنى أن تكون قد أحببت هذه المقالة حول تحليل الطقس والتنبؤ باستخدام بايثون.

2) تحليل وقت الشاشة باستخدام بايثون Screen Time Analysis using Python

يتيح لك تحليل وقت الشاشة (Screen Time Analysis) معرفة مقدار الوقت الذي تقضيهفي نوع التطبيقات والمواقع الإلكترونية التي تستخدم جهازك. ويعطي تحليل وقت الشاشة تقريرًا مرئيًا عن ذلك. لذلك، إذا كنت تريد معرفة كيفية تحليل وقت الشاشة، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة تحليل وقت الشاشة باستخدام بايثون.

تحليل وقت الشاشة

تحليل وقت الشاشة هو مهمة تحليل وإنشاء تقرير عن التطبيقات والمواقع التي يستخدمها المستخدم عن مقدار الوقت. تتمتع أجهزة Apple بواحدة من أفضل الطرق لإنشاء تقرير وقت الشاشة.



بالنسبة لمهمة تحليل وقت الشاشة، وجدت مجموعة بيانات مثالية تحتوي على بيانات حول:

- التاريخ.
- استخدام التطبيقات.
- عدد الاشعارات من التطبيقات.
- عدد مرات فتح التطبيقات.

يمكنك تنزيل مجموعة البيانات من هنا.

في القسم أدناه، سوف آخذك خلال مهمة تحليل وقت الشاشة باستخدام بايثون.

تحليل وقت الشاشة باستخدام بايثون

لنبدأ مهمة تحليل وقت الشاشة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("Screentime - App Details.csv")
print(data.head())
```

```
         Date  Usage  Notifications  Times opened        App
0  08/26/2022     38             70            49  Instagram
1  08/27/2022     39             43            48  Instagram
2  08/28/2022     64            231            55  Instagram
3  08/29/2022     14             35            23  Instagram
4  08/30/2022      3             19             5  Instagram
```

دعنا الآن نلقي نظرة على ما إذا كانت مجموعة البيانات تحتوي على أي قيم فارغة (null values) أم لا:

```
data.isnull().sum()
```

```
Date             0
Usage            0
Notifications    0
Times opened     0
App              0
dtype: int64
```

لا تحتوي مجموعة البيانات على أي قيم خالية. دعنا الآن نلقي نظرة على الإحصائيات الوصفية للبيانات:

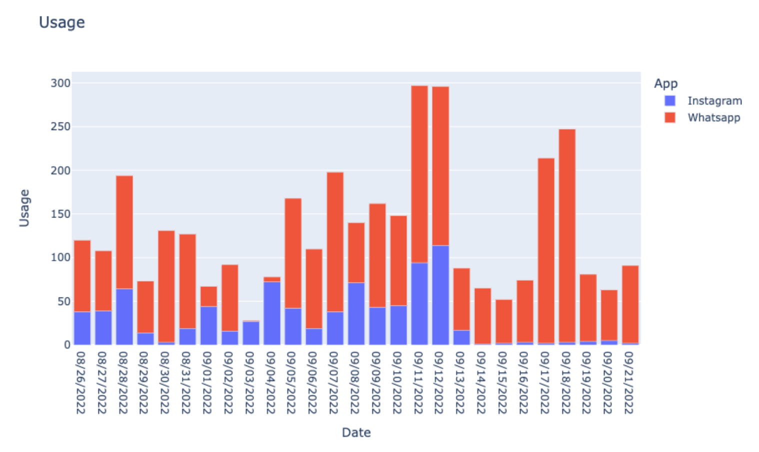
```
print(data.describe())
```

```
            Usage  Notifications  Times opened
count   54.000000      54.000000     54.000000
mean    65.037037     117.703704     61.481481
std     58.317272      97.017530     43.836635
min      1.000000       8.000000      2.000000
25%     17.500000      25.750000     23.500000
50%     58.500000      99.000000     62.500000
75%     90.500000     188.250000     90.000000
max    244.000000     405.000000    192.000000
```

لنبدأ الآن بتحليل وقت الشاشة للمستخدم. سأنظر أولاًفي مقدار استخدام التطبيقات:

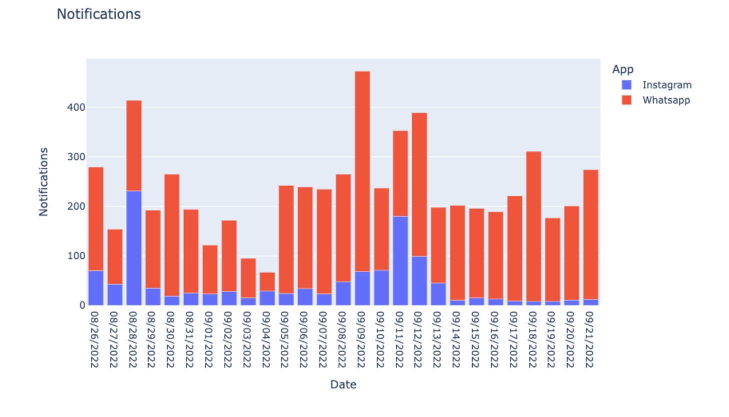
```
figure = px.bar(data_frame=data ,
                x = "Date ",
                y = "Usage ",
                color="App ",
```

```
                    title="Usage ")
figure.show()
```



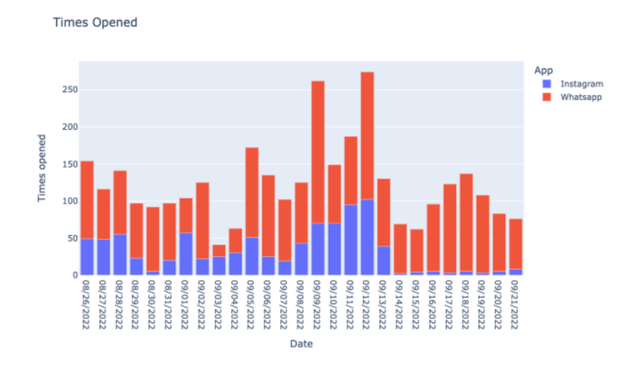
دعنا الآن نلقي نظرة على عدد الإشعارات من التطبيقات:

```
figure = px.bar(data_frame=data ,
                    x = "Date ",
                    y = "Notifications ",
                    color="App ",
                    title="Notifications ")
figure.show()
```



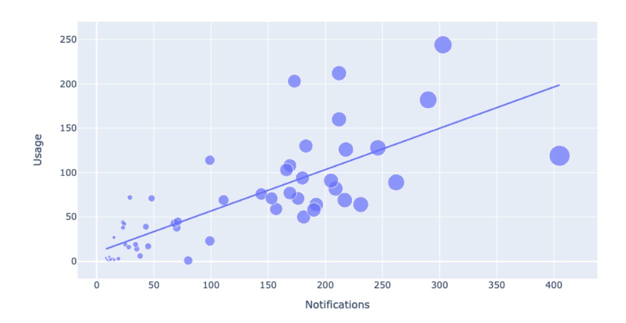
دعنا الآن نلقي نظرة على عدد مرات فتح التطبيقات:

```
figure = px.bar(data_frame=data ,
                    x = "Date ",
                    y = "Times opened ",
                    color="App",
                    title="Times Opened ")
figure.show()
```



نستخدم بشكل عام هواتفنا الذكية عندما نتلقى إشعارًا من أي تطبيق. فلنلقِ نظرة على العلاقة بين عدد الإشعارات ومقدار الاستخدام:

```
figure = px.scatter(data_frame = data ,
                    x="Notifications,"
                    y="Usage ,"
                    size="Notifications ,"
                    trendline="ols ,"
                    title = "Relationship Between Number of
Notifications and Usage("
figure.show()
```



هناك علاقة خطية بين عدد الإشعارات ومقدار الاستخدام. هذا يعني أن المزيد من الإشعارات يؤدي إلى زيادة استخدام الهواتف الذكية.

الملخص

هذه هي الطريقة التي يمكننا بها تحليل وقت الشاشة للمستخدم باستخدام لغة برمجة بايثون. تحليل وقت الشاشة هو مهمة تحليل وإنشاء تقرير عن التطبيقات والمواقع التي يستخدمها المستخدم عن مقدار الوقت. آمل أن تكون قد أحببت هذه المقالة حول تحليل وقت الشاشة باستخدام بايثون.

3) تحليل سوق الأسهم باستخدام بايثون Stock Market Analysis using Python

يعني تحليل سوق الأسهم (Stock Market Analysis) تحليل الاتجاهات الحالية والتاريخية في سوق الأوراق المالية لاتخاذ قرارات البيع والشراء المستقبلية. يعد تحليل سوق الأوراق المالية أحد أفضل حالات استخدام علم البيانات في التمويل. لذا، إذا كنت تريد تعلم كيفية تحليل سوق الأوراق المالية، فهذه المقالة مناسبة لك. في هذه المقالة، سأطلعك على مهمة تحليل سوق الأسهم باستخدام بايثون.

تحليل سوق الأسهم باستخدام بايثون

لتحليل سوق الأسهم، سأجمع بيانات أسعار أسهم Google. في نهاية هذه المقالة، ستتعلم تحليل سوق الأوراق المالية بشكل تفاعلي باستخدام لغة برمجة بايثون. لنبدأ بجمع بيانات أسعار أسهم Google. سأستخدم yfinance API من Yahoo Finance لجمع بيانات أسعار الأسهم. يمكنك معرفة المزيد عن API هنا.

إليك الآن كيفية جمع بيانات أسعار أسهم Google:

```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
import plotly.graph_objects as go
import plotly.express as px

today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=365)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('GOOG',
                      start=start_date,
                      end=end_date,
                      progress=False)
data["Date"] = data.index
data = data[["Date", "Open", "High", "Low",
             "Close", "Adj Close", "Volume"]]
data.reset_index(drop=True, inplace=True)
print(data.head())
```

```
         Date         Open         High          Low        Close    Adj Close  \
0 2021-07-12  2596.669922  2615.399902  2592.000000  2611.280029  2611.280029
1 2021-07-13  2617.629883  2640.840088  2612.739990  2619.889893  2619.889893
2 2021-07-14  2638.030029  2659.919922  2637.959961  2641.649902  2641.649902
3 2021-07-15  2650.000000  2651.899902  2611.959961  2625.330078  2625.330078
4 2021-07-16  2632.820068  2643.659912  2616.429932  2636.909912  2636.909912

   Volume
0  847200
1  830900
2  895600
3  829300
4  742800
```

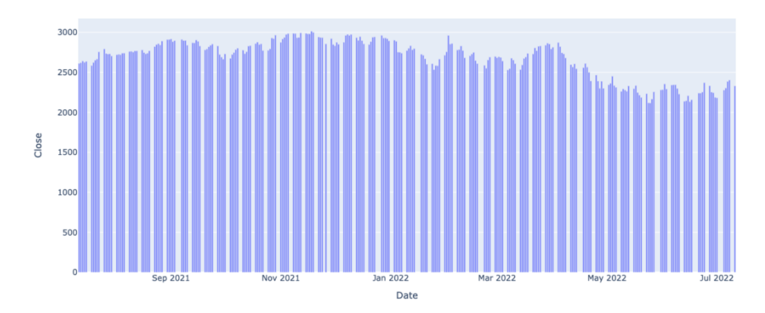
كلما قمت بتحليل سوق الأسهم، ابدأ دائمًا بمخطط الشموع (candlestick chart). مخطط الشموع هو أداة مفيدة لتحليل تحركات الاسعار لأسعار الأسهم. إليك كيفية تصوير مخطط الشموع لأسعار أسهم Google:

```
figure = go.Figure(data=[go.Candlestick(x=data["Date"],
                                        open=data["Open"],
high=data["High"],
                                        low=data["Low"],
close=data["Close"])])
figure.update_layout(title = "Google Stock Price Analysis",
xaxis_rangeslider_visible=False)
figure.show()
```



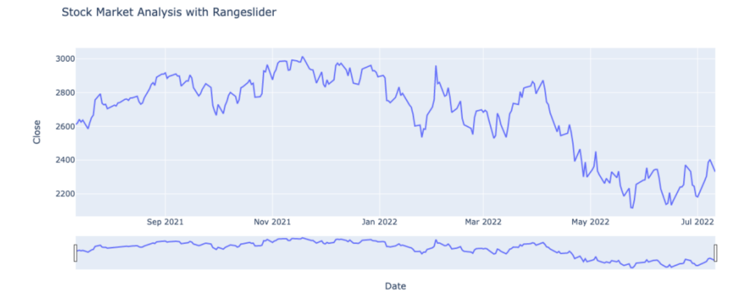
المخطط الشريطي (bar plot) هو أيضًا تصوير مفيد لتحليل سوق الأوراق المالية، على وجه التحديد على المدى الطويل. إليك كيفية تصوير أسعار إغلاق أسهم Google باستخدام مخطط شريطي:

```
figure = px.bar(data, x = "Date", y= "Close")
figure.show()
```



يعد شريط تمرير النطاق (range slider) أحد الأدوات القيمة لتحليل سوق الأوراق المالية. يساعدك على تحليل سوق الأوراق المالية بين نقطتين محددتين من خلال تحديد الفترة الزمنية بشكل تفاعلي. إليك كيفية إضافة شريط تمرير النطاق لتحليل سوق الأسهم:

```
figure = px.line(data, x='Date', y='Close',
                 title='Stock Market Analysis with
Rangeslider')
figure.update_xaxes(rangeslider_visible=True)
figure.show()
```



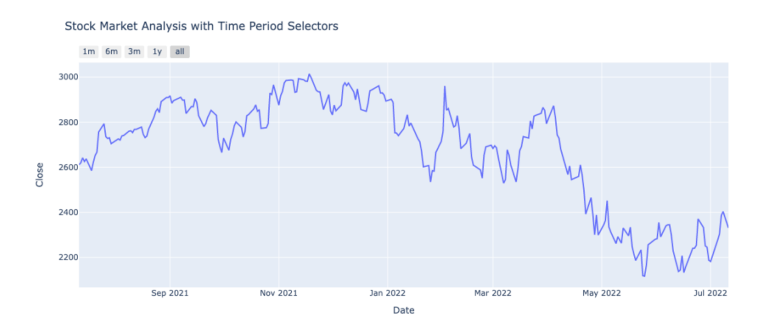
استخدم شريط تمرير النطاق لتحليل سوق الأوراق المالية بشكل تفاعلي بين نقطتين

ميزة تفاعلية أخرى يمكنك إضافتها لتحليل سوق الأسهم هي محددات الفترة الزمنية (time period selectors). محددات الفترة الزمنية هي مثل الأزرار التي تعرض لك الرسم البياني لفترة زمنية محددة. على سبيل المثال، سنة، ثلاثة أشهر، ستة أشهر، إلخ. إليك كيفية إضافة أزرار لاختيار الفترة الزمنية لتحليل سوق الأسهم:

```
figure = px.line(data, x='Date', y='Close ,'
                 title='Stock Market Analysis with Time Period
Selectors('

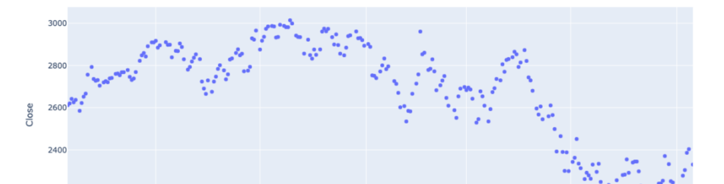
figure.update_xaxes)
    rangeselector=dict)
        buttons=list])
            dict(count=1, label="1m", step="month",
stepmode="backward"),
```

```
                dict(count=6, label="6m", step="month",
stepmode="backward"),
                dict(count=3, label="3m", step="month",
stepmode="backward"),
                dict(count=1, label="1y", step="year",
stepmode="backward"),
                dict(step="all")
([
(
(
figure.show()
```



تؤثر عطلة نهاية الأسبوع أو موسم العطلات دائمًا على سوق الأوراق المالية. لذلك إذاكنت ترغب في إزالة جميع سجلات اتجاهات عطلة نهاية الأسبوع من تصوير سوق الأسهم الخاص بك، فيما يلي كيف يمكنك القيام بذلك:

```
figure = px.scatter(data, x='Date', y='Close', range_x=['2021-
07-12', '2022-07-11,['
                 title="Stock Market Analysis by Hiding
Weekend Gaps("
figure.update_xaxes)
    rangebreaks]=
        dict(bounds=["sat", "sun"])
[
(
figure.show()
```



هذه هي الطريقة التي يمكنك بها تحليل سوق الأسهم باستخدام بايثون. إذا كنت تريد معرفة كيفية التنبؤ بسوق الأوراق المالية، فيمكنك التعلم هنا.

الملخص

هذه هي الطريقة التي يمكنك بها استخدام لغة برمجة بايثون لتحليل سوق الأوراق المالية بشكل تفاعلي. يعني تحليل سوق الأسهم تحليل الاتجاهات الحالية والتاريخية في سوق الأوراق المالية لاتخاذ قرارات البيع والشراء المستقبلية. آمل أن تكون قد أحببت هذه المقالة حول تحليل سوق الأسهم باستخدام بايثون.

4) التنبؤ بالأعمال التجارية باستخدام بايثون Business Forecasting using Python

يعد التنبؤ بالأعمال التجارية (Business Forecasting) أحد تطبيقات التنبؤ بالسلاسل الزمنية (Time Series Forecasting).في تنبؤات الأعمال التجارية، نهدف إلى التنبؤ بالمبيعات أو النفقات أو الإيرادات المستقبلية باستخدام بيانات السلاسل الزمنية التاريخية التي تم إنشاؤها بواسطة الشركة. لذا، إذاكنت تريد معرفة كيفية إجراء التنبؤ بالأعمال التجارية، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بالأعمال باستخدام بايثون.

لماذا يحتاج العمل التجاري إلى التنبؤ بالأعمال التجارية؟

تبحث كل شركة عن استراتيجيات لتحسين أرباحها. يلعب متخصصو علم البيانات دورًا رئيسيًا في توفير أكثر التنبؤات دقةفي أي وقت. دائمًا ما تكون البيانات التي يتم إنشاؤها من قبل الشركة في متناول اليد لتحليل السلوك المستقبلي للعملاء المستهدفين. من خلال التنبؤ باتجاهات الأعمال المستقبلية، يمكن للشركة اتخاذ قرارات أفضل لتحسين أدائهافي المستقبل.

آمل أن تكون قد فهمت سبب احتياج الشركة اليوم إلى استخدام تقنيات التنبؤ بالأعمال. التنبؤ بالمبيعات أو الإيرادات أو النفقات هي بعض حالات استخدام التنبؤ بالأعمال التجارية. لذلك، في القسم أدناه، سوف آخذك خلال مهمة التنبؤ بالأعمال التجارية حيث سنهدف إلى توقع الإيرادات الفصلية لشركة Adidas. يتم جمع البيانات التي أستخدمها لهذه المهمة يدويًا من تقارير المبيعات ربع السنوية من Adidas. يمكنك تنزيل مجموعة البيانات من هنا.

التنبؤ بالأعمال التجارية باستخدام بايثون

لنبدأ بمهمة التنبؤ بالأعمال التجارية عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
from datetime import date, timedelta
import datetime
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.api as sm
import warnings

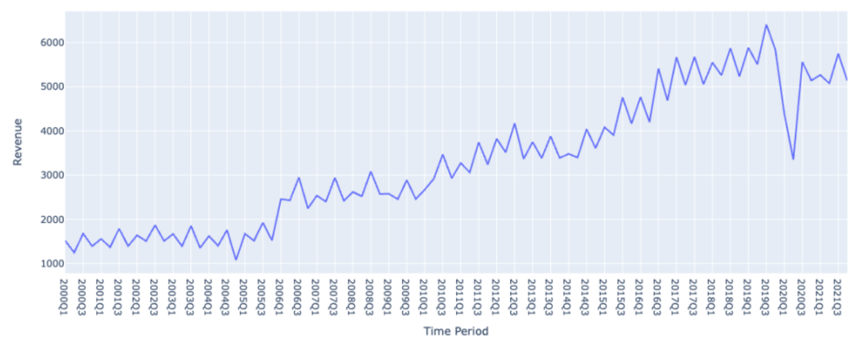
data = pd.read_csv("adidas quarterly sales.csv")
print(data)
```

```
   Time Period  Revenue
0       2000Q1     1517
1       2000Q2     1248
2       2000Q3     1677
3       2000Q4     1393
4       2001Q1     1558
..         ...      ...
83      2020Q4     5142
84      2021Q1     5268
85      2021Q2     5077
86      2021Q3     5752
87      2021Q4     5137

[88 rows x 2 columns]
```

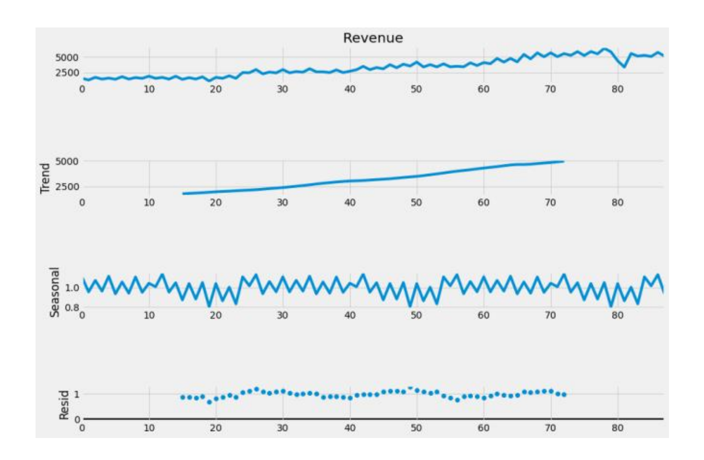
تحتوي مجموعة البيانات على عمودين؛ الفترة الزمنية (Time Period) والإيرادات (Revenue). يحتوي عمود الفترة الزمنية على الإيرادات ربع السنوية لشركة Adidas من 2000 إلى 2021، ويحتوي عمود الإيرادات على إيرادات المبيعات بالملايين (باليورو). دعونا نلقي نظرة على عائدات المبيعات ربع السنوية لشركة Adidas:

```
import plotly.express as px
figure = px.line(data, x="Time Period",
                 y="Revenue",
                 title='Quarterly Sales Revenue of Adidas in
Millions')
figure.show()
```



تعد بيانات إيرادات المبيعات الخاصة بشركة Adidas موسمية حيث تزيد الإيرادات الفصلية وتنخفض كل ربع سنة. فيما يلي كيفية التحقق من موسمية أي بيانات سلاسل زمنية:

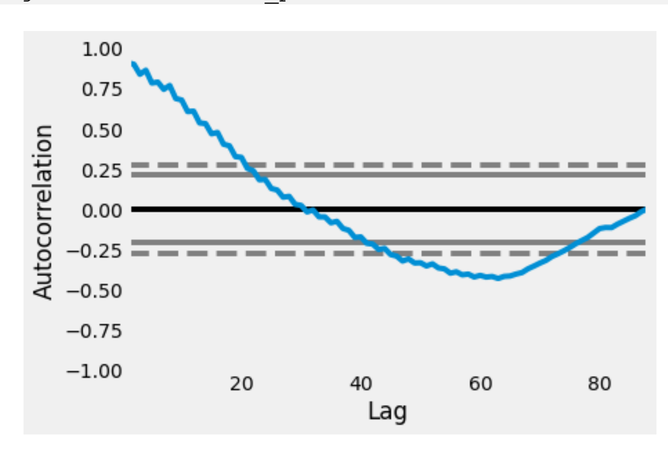
```
result = seasonal_decompose(data["Revenue ,["
                            model='multiplicative', freq = 30(
fig = plt.figure  ()
fig = result.plot  ()
fig.set_size_inches(10 ,15)
```



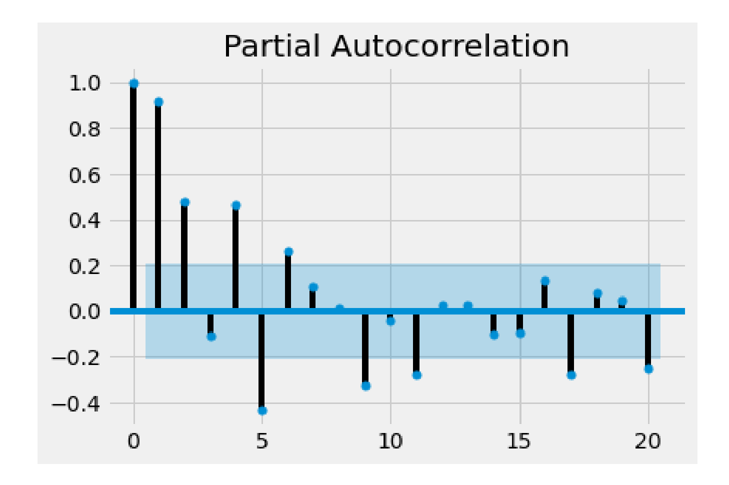
سأستخدم نموذج (SARIMA) Seasonal ARIMA للتنبؤ بإيرادات المبيعات ربع السنوية لشركة Adidas. قبل استخدام نموذج SARIMA، من الضروري إيجاد قيم p و d و q. يمكنك معرفة كيفية العثور على قيم p و d و q من هنا.

نظرًا لأن البيانات ليست ثابتة (not stationary)، فإن قيمة d هي 1. للعثور على قيم p و q، يمكننا استخدام مخططات الارتباط التلقائي (autocorrelation) والارتباط التلقائي الجزئي (partial autocorrelation plots):

```
pd.plotting.autocorrelation_plot(data["Revenue"])
```




```
plot_pacf(data["Revenue"], lags = 20)
```



الآن إليك كيفية تدريب نموذج SARIMA للتنبؤ بالإيرادات ربع السنوية لشركة Adidas:

```
model=sm.tsa.statespace.SARIMAX(data['Revenue"],
                                order=(p, d, q),
                                seasonal_order=(p, d, q, 12))
model=model.fit()
print(model.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                            Revenue   No. Observations:                   88
Model:             SARIMAX(5, 1, 2)x(5, 1, 2, 12)   Log Likelihood                -548.520
Date:                            Mon, 05 Sep 2022   AIC                           1127.041
Time:                                    07:45:33   BIC                           1161.803
Sample:                                         0   HQIC                          1140.921
                                             - 88
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -1.5796      0.391     -4.044      0.000      -2.345      -0.814
ar.L2         -1.4321      0.587     -2.438      0.015      -2.583      -0.281
ar.L3         -0.8305      0.626     -1.328      0.184      -2.057       0.396
ar.L4         -0.5179      0.821     -0.630      0.528      -2.128       1.092
ar.L5         -0.2655      0.491     -0.541      0.589      -1.228       0.697
ma.L1          1.5056      0.518      2.906      0.004       0.490       2.521
ma.L2          0.9697      0.623      1.557      0.120      -0.251       2.190
ar.S.L12      -1.1270    362.141     -0.003      0.998    -710.910     708.656
ar.S.L24      -1.3418    312.728     -0.004      0.997    -614.277     611.594
ar.S.L36      -0.7832    174.955     -0.004      0.996    -343.688     342.122
ar.S.L48      -0.1847     50.633     -0.004      0.997     -99.423      99.054
ar.S.L60      -0.0098      8.921     -0.001      0.999     -17.496      17.476
ma.S.L12       0.3046    362.082      0.001      0.999    -709.363     709.972
ma.S.L24       0.8602    221.641      0.004      0.997    -433.548     435.269
sigma2      1.909e+05   4.01e+05      0.476      0.634   -5.96e+05    9.78e+05
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):               427.98
Prob(Q):                              0.96   Prob(JB):                         0.00
Heteroskedasticity (H):               7.35   Skew:                            -2.04
Prob(H) (two-sided):                  0.00   Kurtosis:                        13.97
===================================================================================
```

الآن دعونا نتوقع الإيرادات ربع السنوية لشركة Adidas للأرباع الثمانية القادمة:

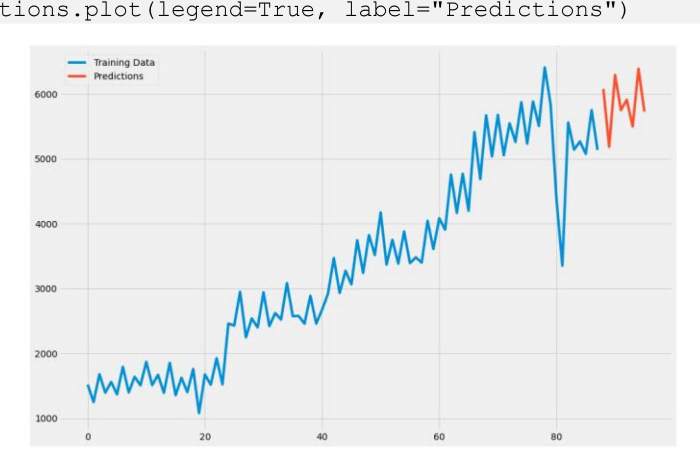
```
predictions = model.predict(len(data), len(data)+7)
```

```
print(predictions)
```

```
88    6078.793918
89    5186.311373
90    6293.196600
91    5751.905629
92    5911.946881
93    5499.784229
94    6389.627988
95    5728.806969
Name: predicted_mean, dtype: float64
```

إليك كيف يمكننا رسم التنبؤات:

```
data["Revenue"].plot(legend=True ,
                     label="Training Data ,"
                     figsize=(15, 10((
predictions.plot(legend=True, label="Predictions")
```



الملخص

هذه هي الطريقة التي يمكنك بها إجراء التنبؤ بالأعمال التجارية باستخدام لغة برمجة بايثون.في تنبؤات الأعمال التجارية، نهدف إلى التنبؤ بالمبيعات أو النفقات أو الإيرادات المستقبلية باستخدام بيانات السلاسل الزمنية التاريخية التي تم إنشاؤها بواسطة الشركة. أتمنى أن تكون قد أحببت هذه المقالة حول التنبؤ بالأعمال التجارية باستخدام بايثون.

5) نظام توصية الأخبار باستخدام بايثون News Recommendation System using Python

نظام التوصية (recommendation system) هو تطبيق شائع لعلم البيانات. تستخدم جميع مواقع الويب الشهيرة التي تزورها تقريبًا أنظمة التوصية. كما يوحي الاسم، فإن نظام التوصية بالأخبار (news recommendation system) هو تطبيق يوصي بالمقالات الإخبارية بناءً على الأخبار التي يقرأها المستخدم بالفعل. لذلك، إذا كنت تريد معرفة كيفية إنشاء نظام توصيات الأخبار، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على كيفية إنشاء نظام توصيات أخبار باستخدام بايثون.

كيف يعمل نظام توصيات الأخبار؟

عندما تزور أي موقع ويب، فإنه يوصي بمحتوى مشابه بناءً على ما تشاهده أو تقرأه بالفعل. تعتبر توصية المحتوى بناءً على المحتوى الذي يستهلكه المستخدم بالفعل تقنية لإنشاء نظام توصية يُعرف باسم التصفية القائمة على المحتوى (Content-based filtering).

تستخدم جميع مواقع الويب الإخبارية الشهيرة أنظمة توصية قائمة على المحتوى مصممة للعثور على أوجه التشابه بين الأخبار التي تقرأها والمقالات الإخبارية الأخرى على موقعها على الويب للتوصية بالمقالات الإخبارية الأكثر تشابهًا.

أتمنى أن تكون قد فهمت الآن كيف يعمل نظام توصية الأخبار.في القسم أدناه، سوف أطلعك على كيفية إنشاء نظام توصيات الأخبار باستخدام لغة برمجة بايثون.

نظام توصية الأخبار باستخدام بايثون

مجموعة البيانات التي أستخدمها لإنشاء نظام توصيات الأخبار مأخوذة من Microsoft. نظرًا لأن البيانات كانت بحاجة إلى الكثير من التنظيف والإعداد، فقد قمت بتنزيل البيانات وأعدتها لإنشاء نظام توصية قائم على المحتوى. يمكنك تنزيل مجموعة البيانات هنا (يُرجى تنزيل البيانات بتنسيق CSV).

لنبدأ الآن باستيراد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لبناء نظام توصية بالأخبار:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import cosine_similarity
import plotly.express as px
import plotly.graph_objects as go
```

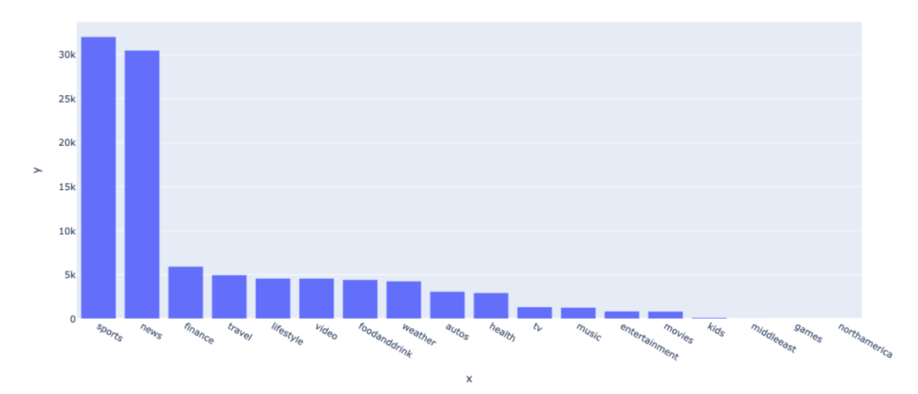
```
data = pd.read_csv("News.csv")
print(data.head())
```

```
       ID News Category                                              Title  \
0  N88753     lifestyle  The Brands Queen Elizabeth, Prince Charles, an...
1  N45436          news    Walmart Slashes Prices on Last-Generation iPads
2  N23144        health                        50 Worst Habits For Belly Fat
3  N86255        health  Dispose of unwanted prescription drugs during ...
4  N93187          news  The Cost of Trump's Aid Freeze in the Trenches...

                                          Summary
0  Shop the notebooks, jackets, and more that the...
1  Apple's new iPad releases bring big deals on l...
2  These seemingly harmless habits are holding yo...
3                                                NaN
4  Lt. Ivan Molchanets peeked over a parapet of s...
```

دعونا نلقي نظرة على فئات الأخبارفي مجموعة البيانات هذه:

```
# Types of News Categories
categories = data["News Category"].value_counts()
label = categories.index
counts = categories.values
figure = px.bar(data, x=label,
                y = counts,
            title="Types of News Categories")
figure.show()
```



توجد طريقتان لإنشاء نظام توصية باستخدام مجموعة البيانات هذه:

1. إذا اخترنا عمود فئة الأخبار (News Category column) كميزة سنستخدمها للعثور على أوجه التشابه، فقد لا تساعد التوصياتفي جذب انتباه المستخدم لفترة أطول. لنفترض أن مستخدمًا يقرأ أخبارًا عن الرياضة بناءً على مباراة كريكيت ويحصل على توصيات إخبارية حول رياضات أخرى مثل المصارعة والهوكي وكرة القدم وما إلى ذلك، والتي قد تكون غير مناسبة وفقًا للمحتوى الذي يقرأه المستخدم.

2. الطريقة الأخرى هي استخدام العنوان أو الملخص كميزة للعثور على أوجه التشابه. ستقدم توصيات أكثر دقة لأن المحتوى الموصى به سيعتمد على المحتوى الذي يقرأه المستخدم بالفعل.

لذلك يمكننا استخدام العنوان (title) أو ملخص (summary) المقال الإخباري للعثور على أوجه التشابه مع المقالات الإخبارية الأخرى. هنا سأستخدم عمود العنوان. إذا كنت ترغب في استخدام عمود الملخص، فقم أولاً بإسقاط الصفوف ذات القيم الخالية (null values)، حيث يحتوي عمود الملخص على أكثر من 5000 قيمة فارغة.

فيما يلي كيف يمكننا إيجاد أوجه التشابه بين المقالات الإخبارية عن طريق تحويل نصوص عمود العنوان إلى متجهات رقمية ثم إيجاد أوجه التشابه بين المتجهات العددية باستخدام خوارزمية تشابه جيب التمام (cosine similarity algorithm):

```
feature = data["Title"].tolist()
tfidf = text.TfidfVectorizer(input=feature,
stop_words="english")
tfidf_matrix = tfidf.fit_transform(feature)
similarity = cosine_similarity(tfidf_matrix)
```

الآن سأقوم بتعيين عمود العنوان كفهرس للبيانات حتى نتمكن من البحث عن توصيات المحتوى من خلال إعطاء العنوان كمدخل:

```
indices = pd.Series(data.index,
index=data['Title']).drop_duplicates()
```

الآن فيما يلي كيفية إنشاء نظام توصيات الأخبار:

```
def news_recommendation(Title, similarity = similarity):
    index = indices[Title]
    similarity_scores = list(enumerate(similarity[index]))
    similarity_scores = sorted(similarity_scores,
    key=lambda x: x[1], reverse=True)
    similarity_scores = similarity_scores[0:10]
    newsindices = [i[0] for i in similarity_scores]
    return data['Title'].iloc[newsindices]

print(news_recommendation("Walmart Slashes Prices on Last-
Generation iPads"))
```

```
1           Walmart Slashes Prices on Last-Generation iPads
83827     Walmart's Black Friday 2019 ad: the best deals...
76024     Walmart Black Friday 2019 deals unveiled: Huge...
90316     US consumer prices up 0.4% in October; gasolin...
89588     Consumer prices rise most in 7 months on highe...
32839                 Inside the next generation of irons
37970     Walmart and Kroger Undercut Drugstore Chains' ...
100684    Nissan slashes full-year forecast as first-hal...
74916                  The Top Deals at Walmart Right Now
39634     Federal Reserve slashes interest rates for thi...
Name: Title, dtype: object
```

هذه هي الطريقة التي يمكنك بها إنشاء نظام توصية بالأخبار باستخدام لغة برمجة بايثون.

الملخص

تستخدم جميع مواقع الويب الإخبارية الشهيرة أنظمة توصية قائمة على المحتوى مصممة للعثور على أوجه التشابه بين الأخبار التي تقرأها والمقالات الإخبارية الأخرى على موقعها على الويب للتوصية بالمقالات الإخبارية الأكثر تشابهًا. أتمنى أن تكون قد أحببت هذه المقالة حول كيفية إنشاء نظام توصية بالأخبار باستخدام بايثون.

6) تحليل مبيعات iPhone باستخدام بايثون iPhone Sales Analysis using Python

تعد أجهزة Apple iPhones من بين الهواتف الذكية الأكثر مبيعًافي جميع أنحاء العالم. هناك منافسة كبيرة بين العلامات التجارية للهواتف الذكيةفي الهند، حيث يمكنك الحصول على أحدث التقنياتفي هاتف ذكي بنصف سعر iPhone. لا تزال هناك مبيعات عالية من أجهزة iPhone في الهند. لذلك إذا كنت ترغبفي تحليل مبيعات iPhoneفي الهند، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل مبيعات iPhone باستخدام بايثون.

تحليل مبيعات iPhone باستخدام بايثون

بالنسبة لمهمة تحليل مبيعات iPhone، قمت بجمع مجموعة بيانات من Kaggle تحتوي على بيانات حول مبيعات أجهزة iPhoneفي الهند على Flipkart. ستكون مجموعة بيانات مثالية لتحليل مبيعات iPhoneفي الهند. يمكنك تنزيل مجموعة البيانات من هنا.

دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة لبدء مهمة تحليل مبيعات iPhone:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("apple_products.csv")
print(data.head())
```

```
                          Product Name  \
0          APPLE iPhone 8 Plus (Gold, 64 GB)
1  APPLE iPhone 8 Plus (Space Grey, 256 GB)
2      APPLE iPhone 8 Plus (Silver, 256 GB)
3           APPLE iPhone 8 (Silver, 256 GB)
4             APPLE iPhone 8 (Gold, 256 GB)

                                         Product URL  Brand  Sale Price  \
0  https://www.flipkart.com/apple-iphone-8-plus-g...  Apple       49900
1  https://www.flipkart.com/apple-iphone-8-plus-s...  Apple       84900
2  https://www.flipkart.com/apple-iphone-8-plus-s...  Apple       84900
3  https://www.flipkart.com/apple-iphone-8-silver...  Apple       77000
4  https://www.flipkart.com/apple-iphone-8-gold-2...  Apple       77000

     Mrp  Discount Percentage  Number Of Ratings  Number Of Reviews  \
0  49900                    0               3431                356
1  84900                    0               3431                356
2  84900                    0               3431                356
3  77000                    0              11202                794
4  77000                    0              11202                794

                Upc  Star Rating   Ram
0  MOBEXRGV7EHHTGUH          4.6  2 GB
1  MOBEXRGVAC6TJT4F          4.6  2 GB
2  MOBEXRGVGETABXWZ          4.6  2 GB
3  MOBEXRGVMZWUHCBA          4.5  2 GB
4  MOBEXRGVPK7PFEJZ          4.5  2 GB
```

قبل المضي قدمًا، دعنا نلقي نظرة سريعة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة (null values) أم لا:

```
print(data.isnull().sum())
```

```
Product Name           0
Product URL            0
Brand                  0
Sale Price             0
Mrp                    0
Discount Percentage    0
Number Of Ratings      0
Number Of Reviews      0
Upc                    0
Star Rating            0
Ram                    0
dtype: int64
```

لا تحتوي مجموعة البيانات على أي قيم خالية. الآن، دعنا نلقي نظرة على الإحصائيات الوصفية للبيانات:

```
print(data.describe())
```

```
          Sale Price            Mrp  Discount Percentage  Number Of Ratings  \
count      62.000000      62.000000            62.000000          62.000000
mean    80073.887097   88058.064516             9.951613       22420.403226
std     34310.446132   34728.825597             7.608079       33768.589550
min     29999.000000   39900.000000             0.000000         542.000000
25%     49900.000000   54900.000000             6.000000         740.000000
50%     75900.000000   79900.000000            10.000000        2101.000000
75%    117100.000000  120950.000000            14.000000       43470.000000
max    140900.000000  149900.000000            29.000000       95909.000000

       Number Of Reviews  Star Rating
count          62.000000    62.000000
mean         1861.677419     4.575806
std          2855.883830     0.059190
min            42.000000     4.500000
25%            64.000000     4.500000
50%           180.000000     4.600000
75%          3331.000000     4.600000
max          8161.000000     4.700000
```

تحليل مبيعات iPhone في الهند

الآن سوف أقوم بإنشاء إطار بيانات جديد من خلال تخزين جميع البيانات حول أعلى 10 أجهزة iPhone تصنيفاً في الهند على Flipkart. سيساعد ذلك في فهم نوع أجهزة iPhone التي تحظى بإعجاب أكبر في الهند:

```
highest_rated = data.sort_values(by=["Star Rating"],
```

```
                                  ascending=False)
highest_rated = highest_rated.head(10)
print(highest_rated['Product Name'])
```

```
20     APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)
17         APPLE iPhone 11 Pro Max (Space Grey, 64 GB)
16    APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)
15               APPLE iPhone 11 Pro Max (Gold, 64 GB)
14              APPLE iPhone 11 Pro Max (Gold, 256 GB)
0                   APPLE iPhone 8 Plus (Gold, 64 GB)
29                    APPLE iPhone 12 (White, 128 GB)
32          APPLE iPhone 12 Pro Max (Graphite, 128 GB)
35                    APPLE iPhone 12 (Black, 128 GB)
36                     APPLE iPhone 12 (Blue, 128 GB)
Name: Product Name, dtype: object
```

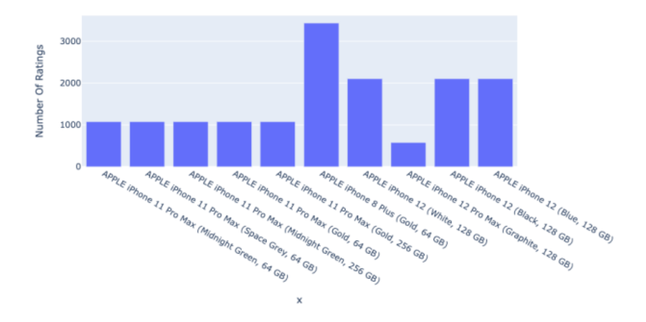
وفقًا للبيانات المذكورة أعلاه، يوجد أدناه أهم 5 أجهزة iPhone الأكثر شهرة في الهند:

1. APPLE iPhone 11 Pro Max (أخضر ليلي ، 64 جيجابايت).
2. APPLE iPhone 11 Pro Max (رمادي فلكي ، 64 جيجابايت).
3. APPLE iPhone 11 Pro Max (أخضر ليلي ، 256 جيجابايت).
4. APPLE iPhone 11 Pro Max (ذهبي ، 64 جيجابايت).
5. APPLE iPhone 11 Pro Max (ذهبي ، 256 جيجابايت).

دعنا الآن نلقي نظرة على عدد تقييمات أجهزة iPhone الأعلى تصنيفًا على Flipkart:

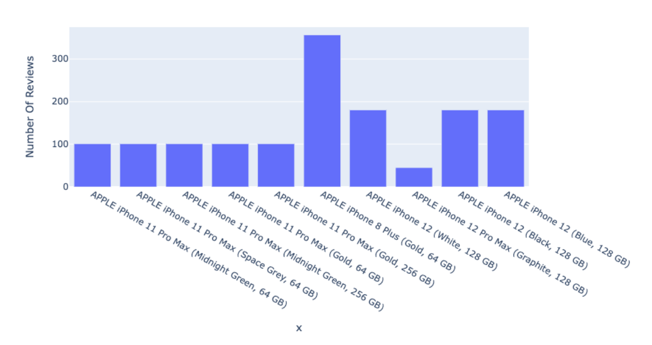
```
iphones = highest_rated["Product Name"].value_counts()
label = iphones.index
counts = highest_rated["Number Of Ratings"]
figure = px.bar(highest_rated, x=label,
                y = counts,
            title="Number of Ratings of Highest Rated
iPhones")
figure.show()
```





وفقًا للرسم البياني الشريطي (bar graph) أعلاه، يتمتع APPLE iPhone 8 Plus (ذهبي، 64 جيجابايت) بأكبر عدد من التقييمات على Flipkart. دعنا الآن نلقي نظرة على عدد المراجعات لأجهزة iPhone الأعلى تقييمًا على Flipkart:

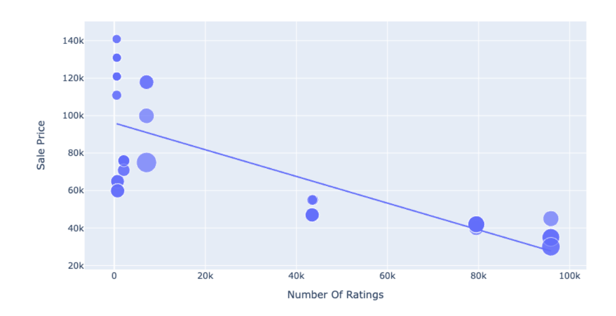
```
iphones = highest_rated["Product Name"].value_counts()
label = iphones.index
counts = highest_rated["Number Of Reviews"]
figure = px.bar(highest_rated, x=label,
                y = counts,
            title="Number of Reviews of Highest Rated
iPhones")
figure.show()
```



يتصدر APPLE iPhone 8 Plus (ذهبي، 64 جيجابايت) أيضًا أعلى عدد من المراجعات على Flipkart من بين أجهزة iPhone الأعلى تصنيفًا في الهند. دعنا الآن نلقي نظرة على العلاقة بين سعر بيع أجهزة iPhone وتقييماتها على Flipkart:

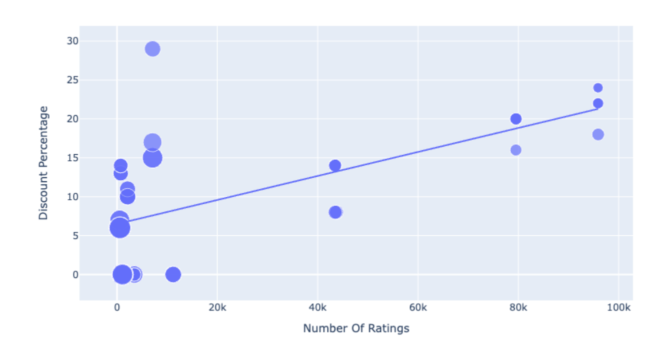
```
figure = px.scatter(data_frame = data, x="Number Of Ratings,"
                    y="Sale Price", size="Discount
Percentage ,"
                    trendline="ols ,"
                    title="Relationship between Sale Price and
Number of Ratings of iPhones("
figure.show()
```





توجد علاقة خطية سالبة بين سعر بيع أجهزة iPhone وعدد التقييمات. هذا يعني أن أجهزة iPhone ذات أسعار البيع المنخفضة تُباع أكثرفي الهند. دعنا الآن نلقي نظرة على العلاقة بين نسبة الخصم على أجهزة iPhone على Flipkart وعدد التقييمات:

```
figure = px.scatter(data_frame = data, x="Number Of Ratings,"
                    y="Discount Percentage", size="Sale
Price ,"
                    trendline="ols ,"
                    title="Relationship between Discount
Percentage and Number of Ratings of iPhones("
figure.show()
```



هناك علاقة خطية بين نسبة الخصم على أجهزة iPhone على Flipkart وعدد التقييمات. هذا يعني أن أجهزة iPhone ذات الخصومات العالية تُباع أكثرفي الهند.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل مبيعات iPhoneفي الهند باستخدام لغة برمجة بايثون. بعض النقاط المستفادة من هذه المقالة حول مبيعات iPhoneفي الهند هي:

- كان APPLE iPhone 8 Plus (ذهبي، 64 جيجابايت) أكثر iPhone تقديراًفي الهند.
- تباع أجهزة iPhone ذات أسعار البيع المنخفضة أكثرفي الهند.
- تباع أجهزة iPhone ذات الخصومات العالية أكثرفي الهند.

أتمنى أن تكون قد أحببت مقالة تحليل مبيعات iPhone باستخدام بايثون.

7) تحليل مبيعات Flipkart باستخدام بايثون Flipkart Sale Analysis using Python

تقدم Flipkart عملية بيع كل عام خلال موسم الأعيادفي الهند. يُعرف هذا البيع باسم بيع Big Billion Days. ينصب تركيز البيع على الإلكترونيات من كل نوع، بمافي ذلك الهواتف الذكية الرائدة. لذلك، إذاكنت ترغب في تحليل المنتجات والعروض على بيع منصة التجارة الإلكترونية، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل مبيعات Flipkart باستخدام بايثون.

تحليل مبيعات Flipkart

لتحليل بيع Flipkart Big Billion Days، بحثت ووجدت مجموعة بيانات مثالية على Kaggle. تحتوي مجموعة البيانات على جميع المعلومات حول جميع المنتجات المعروضة للبيع. نظرًا لبدء هذا البيعفي 23 سبتمبر، يمكننا تحليل اليوم الأول من البيع حيث تم تقديم العروض الأكثر جاذبيةفي اليوم الأول بواسطة Flipkart.

بالنسبة لمهمة تحليل مبيعات Flipkart، سأختار فقط بيانات البيع على الهواتف الذكيةفي اليوم الأول من بيع Flipkart. يمكنك تنزيل مجموعة البيانات التي أستخدمها لهذه المهمة هنا. يمكنك أيضًا استخدام مجموعة البيانات الكاملة لبيع Flipkart هنا.

في القسم أدناه، سأنتقل بك إلى تحليل مبيعات Flipkart من خلال تحليل العروض على بيع الهواتف الذكية.

تحليل مبيعات Flipkart باستخدام بايثون

لنبدأ مهمة تحليل بيع Flipkart عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("23_09_2022.csv")
print(data.head())
```

```
                               name  offer_price  original_price  off_now  \
0      APPLE iPhone 13 (Blue, 128 GB)        57990           69900  17% off
1     APPLE iPhone 11 (White, 128 GB)        41990           48900  14% off
2  APPLE iPhone 13 (Midnight, 128 GB)        57990           69900  17% off
3                            IAIR D25         1098            1699  35% off
4  APPLE iPhone 13 (Starlight, 128 GB)       58990           69900  15% off

   total_ratings  total_reviews  rating  \
0          13052           1036     4.6
1          96244           7044     4.6
2          13052           1036     4.6
3             11              8     4.1
4          13052           1036     4.6

                                         description  \
0  ['128 GB ROM', '15.49 cm (6.1 inch) Super Reti...
1  ['128 GB ROM', '15.49 cm (6.1 inch) Liquid Ret...
2  ['128 GB ROM', '15.49 cm (6.1 inch) Super Reti...
3  ['32 MB RAM | 32 MB ROM', '4.32 cm (1.7 inch) ...
4  ['128 GB ROM', '15.49 cm (6.1 inch) Super Reti...

                        created_at
0  2022-09-23 22:37:42.702432+05:30
1  2022-09-23 22:37:42.703432+05:30
2  2022-09-23 22:37:42.703432+05:30
3  2022-09-23 22:37:42.703432+05:30
4  2022-09-23 22:37:42.704431+05:30
```

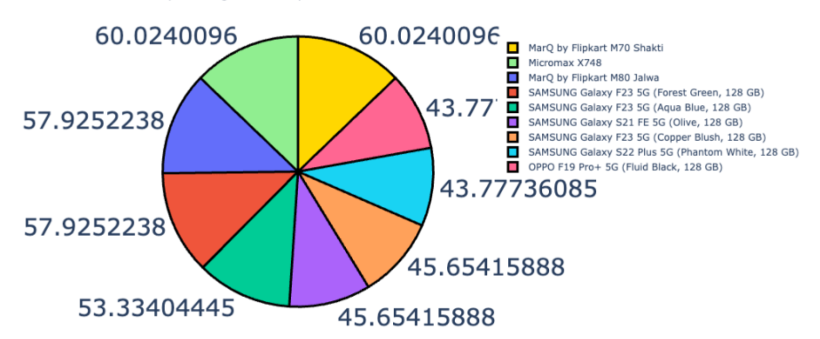
يحتوي عمود الخصم (discount column) المذكورفي مجموعة البيانات على قيم سلسلة. لذلك سوف أقوم بإنشاء عمود خصم جديد من خلال حساب الخصم الذي تقدمه Flipkart على كل هاتف ذكي:

```
data["Discount"] = (data['original_price'] -
data['offer_price']) / data['original_price'] * 100
```

دعنا الآن نلقي نظرة على أفضل الصفقات على الهواتف الذكية التي تقدمها Flipkartفي التخفيضات:

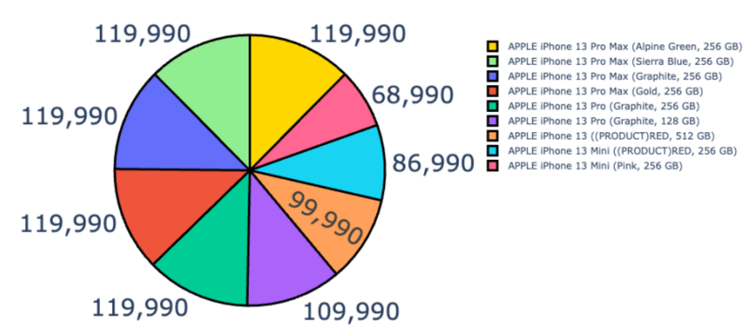
```
top_deals = data.sort_values(by="Discount", ascending=False)
deals = top_deals["name"][:15].value_counts()
label = deals.index
counts = top_deals["Discount"][:15].values
colors = ['gold','lightgreen']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Highest Discount Deals in the
Flipkart Big Billion Days Sale')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
         marker=dict(colors=colors, line=dict(color='black',
width=3))(
fig.show()
```





حتى نتمكن من رؤية المنتجات الأعلى تصنيفًا من جميع شرائح الأسعارفي هذا البيع. دعنا الآن نلقي نظرة على أغلى صفقات الهواتف الذكيةفي التخفيضات:

```
most_expensive = data.sort_values(by="offer_price",
ascending=False)
deals = most_expensive["name"][:10].value_counts()
label = deals.index
counts = most_expensive["offer_price"][:10].values
colors = ['gold','lightgreen']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Most Expensive Offers in the
Flipkart Big Billion Days Sale')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
                  marker=dict(colors=colors,
line=dict(color='black', width=3)))
fig.show()
```



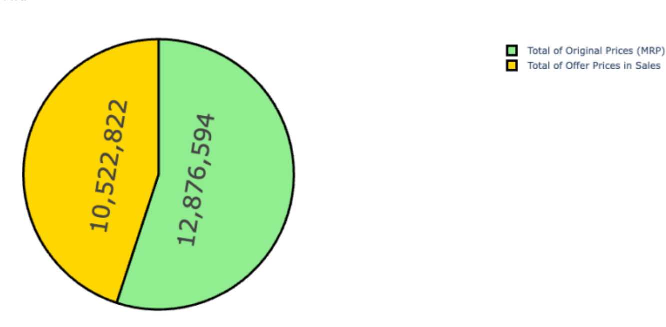
جميع العروض باهظة الثمن على الهواتف الذكيةفي البيع كانت على أجهزة Apple iPhones.

التكلفة اليومية لبيع الهواتف الذكية إلى Flipkart

الخصومات المقدمة هي نفقات الأعمال. يخفض النشاط التجاري سعر المنتج لزيادة بيعه للمنتجات. الخصم يندرج تحت فئة التكاليف الترويجية.

نحن نستخدم بيانات بيع Flipkart على الهواتف الذكيةفي اليوم الأول من البيع. لذلك دعونا نحسب تكلفة هذا البيع إلى Flipkart على الهواتف الذكية فقطفي اليوم الأول من البيع:

```
label = ["Total of Offer Prices in Sales", "Total of Original
Prices (MRP)"]
counts = [sum(data["offer_price"]),
sum(data["original_price"])]
colors = ['gold','lightgreen']
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Total Discounts Offered Vs.
MRP')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3))(
fig.show()
```




```
print("Cost of big billion days sale to flipkart on
smartphones = ", 12876594 - 10522822)
```

```
Cost of big billion days sale to flipkart on smartphones =  2353772
```

لذا فإن تكلفة الخصم على Flipkart على الهواتف الذكية فقط ستكون 23،53،772 دولارًا لكمية واحدة فقط من جميع الهواتف الذكية المعروضة للبيع.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل مبيعات Flipkart Big Billion Days. لا تترددفي الاستلهام من هذه المقالة لتحليل المزيد من بيع Flipkartعلى مجموعة البيانات الكاملة هنا. آمل أن تكون قد أحببت هذه المقالة حول تحليل بيع Flipkart باستخدام بايثون.

8) تحليل أسعار الماس باستخدام بايثون Diamond Price Analysis using Python

الماس (diamond) هو من أغلى الأحجار. يختلف سعر الماس بغض النظر عن الحجم بسبب العوامل التي تؤثر على سعر الماس. لذلك، إذا كنت تريد معرفة كيفية استخدام مهاراتك في علم البيانات لتحليل سعر الماس والتنبؤ به، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل أسعار الماس والتنبؤ باستخدام لغة برمجة بايثون.

تحليل سعر الماس

لتحليل سعر الماس وفقًا لخصائصه، نحتاج أولاً إلى مجموعة بيانات تحتوي على أسعار الماس بناءً على ميزاتها. لقد وجدت بيانات مثالية على Kaggle تحتوي على معلومات حول الماس مثل:

1. القيراط (Carat).
2. القطع (Cut).
3. اللون (Colour).
4. الوضوح (Clarity).
5. العمق (Depth).
6. الجدول (Table).
7. السعر (Price).
8. الحجم (Size).

يمكنك تنزيل مجموعة البيانات هذه من هنا.في القسم أدناه، سوف آخذك خلال مهمة تحليل سعر الماس باستخدام بايثون.

تحليل أسعار الماس باستخدام بايثون

لنبدأ مهمة تحليل أسعار الماس من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("diamonds.csv")
print(data.head())
```

```
   Unnamed: 0  carat      cut color clarity  depth  table  price     x     y  \
0           1   0.23    Ideal     E     SI2   61.5   55.0    326  3.95  3.98
1           2   0.21  Premium     E     SI1   59.8   61.0    326  3.89  3.84
2           3   0.23     Good     E     VS1   56.9   65.0    327  4.05  4.07
3           4   0.29  Premium     I     VS2   62.4   58.0    334  4.20  4.23
4           5   0.31     Good     J     SI2   63.3   58.0    335  4.34  4.35

      z
0  2.43
1  2.31
2  2.31
3  2.63
4  2.75
```

تحتوي مجموعة البيانات هذه على عمود بدون اسم Unnamed column. سأحذف هذا العمود قبل الانتقال إلى أبعد من ذلك:

```
data = data.drop("Unnamed: 0",axis=1)
```

لنبدأ الآنفي تحليل أسعار الماس. سأحلل أولاً العلاقة بين القيراط وسعر الماس لأرى كيف يؤثر عدد القيراط على سعر الماس:

```
figure = px.scatter(data_frame = data, x="carat",
         y="price", size="depth",
         color= "cut", trendline="ols")
figure.show()
```

يمكننا أن نرى علاقة خطية بين عدد قيراط وسعر الماس. هذا يعني أن القيراط الأعلى يؤدي إلى ارتفاع الأسعار.

سأضيف الآن عمودًا جديدًا إلى مجموعة البيانات هذه عن طريق حساب حجم الماس (الطول × العرض × العمق):

```
data["size"] = data["x"] * data["y"] * data["z"]
```

```
print(data)
```

```
       carat        cut color clarity  depth  table  price     x     y     z  \
0       0.23      Ideal     E     SI2   61.5   55.0    326  3.95  3.98  2.43
1       0.21    Premium     E     SI1   59.8   61.0    326  3.89  3.84  2.31
2       0.23       Good     E     VS1   56.9   65.0    327  4.05  4.07  2.31
3       0.29    Premium     I     VS2   62.4   58.0    334  4.20  4.23  2.63
4       0.31       Good     J     SI2   63.3   58.0    335  4.34  4.35  2.75
...      ...        ...   ...     ...    ...    ...    ...   ...   ...   ...
53935   0.72      Ideal     D     SI1   60.8   57.0   2757  5.75  5.76  3.50
53936   0.72       Good     D     SI1   63.1   55.0   2757  5.69  5.75  3.61
53937   0.70  Very Good     D     SI1   62.8   60.0   2757  5.66  5.68  3.56
53938   0.86    Premium     H     SI2   61.0   58.0   2757  6.15  6.12  3.74
53939   0.75      Ideal     D     SI2   62.2   55.0   2757  5.83  5.87  3.64

             size
0       38.202030
1       34.505856
2       38.076885
3       46.724580
4       51.917250
...           ...
53935  115.920000
53936  118.110175
53937  114.449728
53938  140.766120
53939  124.568444

[53940 rows x 11 columns]
```

دعنا الآن نلقي نظرة على العلاقة بين حجم الماس وسعره:

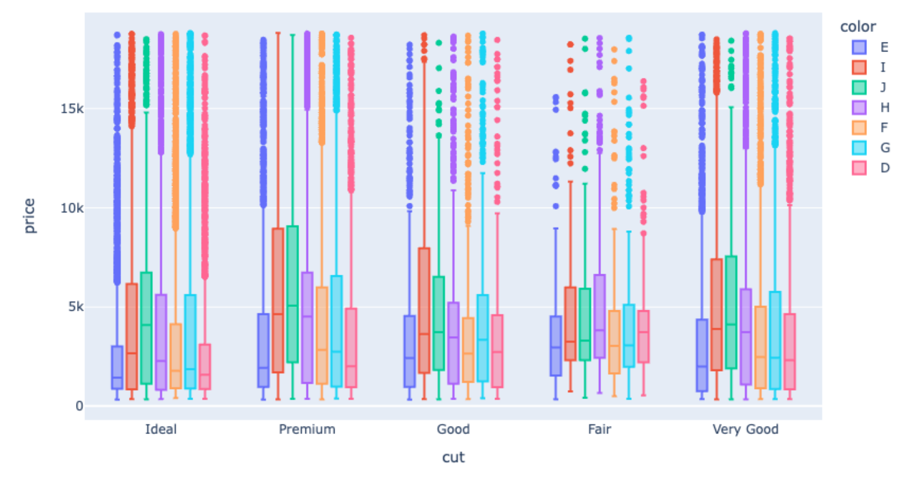
```
figure = px.scatter(data_frame = data, x="size,"
         y="price", size="size,"
         color= "cut", trendline="ols("
figure.show()
```

يستنتج الشكل أعلاه ميزتين للماس:

1. الماس المقطوع المتميز كبير نسبيًا من الماس الآخر.
2. هناك علاقة خطية بين حجم جميع أنواع الماس وأسعاره.

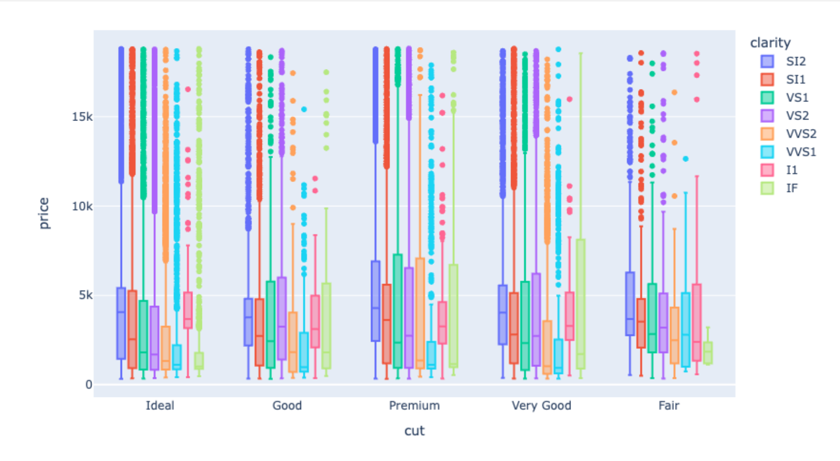
دعنا الآن نلقي نظرة على أسعار جميع أنواع الماس بناءً على لونها:

```
fig = px.box(data, x="cut",
      y="price",
      color="color")
fig.show()
```



دعنا الآن نلقي نظرة على أسعار جميع أنواع الماس بناءً على وضوحها:

```
fig = px.box(data ,
      x="cut",
      y="price",
      color="clarity")
fig.show()
```



دعنا الآن نلقي نظرة على العلاقة بين أسعار الماس والميزات الأخرى في مجموعة البيانات:

```
correlation = data.corr()
print(correlation["price"].sort_values(ascending=False))
```

```
price     1.000000
carat     0.921591
size      0.902385
x         0.884435
y         0.865421
z         0.861249
table     0.127134
depth    -0.010647
Name: price, dtype: float64
```

التنبؤ بسعر الماس

الآن ، سأنتقل إلى مهمة التنبؤ بأسعار الماس باستخدام جميع المعلومات الضرورية من تحليل سعر الماس الذي تم إجراؤه أعلاه.

قبل المضي قدمًا ، سأقوم بتحويل قيم عمود القطع (cut column) حيث أن نوع قطع الماس هو ميزة قيمة للتنبؤ بسعر الماس. لاستخدام هذا العمود ، نحتاج إلى تحويل قيمه الفئوية (categorical values) إلى قيم عددية (numerical values). فيما يلي كيف يمكننا تحويلها إلى ميزة عددية:

```
data["cut"] = data["cut"].map({"Ideal": 1,
               "Premium": 2,
              "Good": 3,
              "Very Good": 4,
              "Fair": 5({
```

الآن، دعنا نقسم البيانات إلى مجموعات تدريب واختبار:

```
#splitting data
from sklearn.model_selection import train_test_split
x = np.array(data[["carat", "cut", "size"]])
y = np.array(data[["price"]])

xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                test_size=0.10,
                                                random_state=42)
```

الآن سأقوم بتدريب نموذج التعلم الآلي لمهمة التنبؤ بسعر الماس:

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor()
model.fit(xtrain, ytrain)
```

الآن فيما يلي كيفية استخدام نموذج التعلم الآلي الخاص بنا للتنبؤ بسعر الماس:

```
print("Enter House Details to Predict Rent")
a = float(input("Carat Size: "))
b = int(input("Cut Type (Ideal: 1, Premium: 2, Good: 3, Very
Good: 4, Fair: 5): "))
c = float(input("Size: "))
features = np.array([[a, b, c]])
print("Predicted Diamond's Price = ", model.predict(features))
```

```
Enter House Details to Predict Rent
Carat Size: 0.60
Cut Type (Ideal: 1, Premium: 2, Good: 3, Very Good: 4, Fair: 5): 2
Size: 40
Predicted Diamond's Price =  [937.13946429]
```

الملخص

هذه هي الطريقة التي يمكنك بها استخدام مهاراتك في علوم البيانات لمهمة تحليل أسعار الماس والتنبؤ باستخدام لغة برمجة بايثون. وفقًا لتحليل أسعار الماس، يمكننا القول إن سعر وحجم الماس المتميز أعلى من أنواع الماس الأخرى. أتمنى أن تكون قد أحببت هذا المقال عن تحليل سعر الماس والتنبؤ به باستخدام بايثون.

9) نظام توصيات Netflix باستخدام بايثون Netflix Recommendation System using Python

Netflix عبارة عن نظام أساسي للبث يعتمد على الاشتراك يسمح للمستخدمين بمشاهدة الأفلام والبرامج التلفزيونية بدون إعلانات. أحد أسباب شعبية Netflix هو نظام التوصية الخاص بها (recommendation system). يوصي نظام التوصية بالأفلام والبرامج التلفزيونية بناءً على اهتمام المستخدم. إذا كنت طالبًا في علوم البيانات وترغب في معرفة كيفية إنشاء نظام توصية Netflix، فهذه المقالة مناسبة لك. ستأخذك هذه المقالة في جولة حول كيفية إنشاء نظام توصية Netflix باستخدام بايثون.

كيفية عمل نظام توصيات Netflix

يعرض لك نظام التوصية في Netflix الأفلام والبرامج التلفزيونية وفقًا لاهتماماتك. Netflix لديها الكثير من البيانات بسبب قاعدة مستخدميها. يتنبأ نظام التوصية الخاص به بكتالوج مخصص لك بناءً على عوامل مثل:

1. سجل المشاهدة الخاص بك.
2. محفوظات المشاهدة للمستخدمين الآخرين الذين لديهم أذواق وتفضيلات مماثلة لذوقك وتفضيلاتك.
3. الأنواع والفئات والوصف والمزيد من المعلومات حول المحتوى الذي شاهدته في الماضي.

يعد نوع المحتوى أحد أهم العوامل التي تساعد Netflix على التوصية بمزيد من المحتوى حتى للمستخدمين الجدد. آمل أن تكون قد فهمت كيف توصي Netflix مستخدميها بالمحتوى. يمكنك معرفة المزيد عن ذلك هنا. في القسم أدناه، سوف أطلعك على كيفية إنشاء نظام توصية Netflix باستخدام بايثون.

نظام توصيات Netflix باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لإنشاء نظام توصية Netflix باستخدام بايثون من Kaggle. تحتوي مجموعة البيانات على معلومات حول جميع الأفلام والبرامج التلفزيونية على Netflix اعتبارًا من عام 2021. يمكنك تنزيل مجموعة البيانات من هنا.

دعنا الآن نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import numpy as np
```

```
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import cosine_similarity

data = pd.read_csv("netflixData.csv")
print(data.head())
```

```
                                Show Id                          Title  \
0  cc1b6ed9-cf9e-4057-8303-34577fb54477                       (Un)Well
1  e2ef4e91-fb25-42ab-b485-be8e3b23dedb                         #Alive
2  b01b73b7-81f6-47a7-86d8-acb63080d525  #AnneFrank - Parallel Stories
3  b6611af0-f53c-4a08-9ffa-9716dc57eb9c                       #blackAF
4  7f2d4170-bab8-4d75-adc2-197f7124c070               #cats_the_mewvie

                                         Description  \
0  This docuseries takes a deep dive into the luc...
1  As a grisly virus rampages a city, a lone man ...
2  Through her diary, Anne Frank's story is retol...
3  Kenya Barris and his family navigate relations...
4  This pawesome documentary explores how our fel...

                      Director  \
0                          NaN
1                       Cho Il
2  Sabina Fedeli, Anna Migotto
3                          NaN
4             Michael Margolis

                                         Genres  \
0                                     Reality TV
1  Horror Movies, International Movies, Thrillers
2             Documentaries, International Movies
3                                     TV Comedies
4             Documentaries, International Movies
                                                Cast Production Country  \
0                                                NaN      United States
1                           Yoo Ah-in, Park Shin-hye        South Korea
2                         Helen Mirren, Gengher Gatti              Italy
3  Kenya Barris, Rashida Jones, Iman Benson, Genn...      United States
4                                                NaN             Canada

   Release Date Rating  Duration Imdb Score Content Type         Date Added
0        2020.0  TV-MA  1 Season     6.6/10      TV Show                NaN
1        2020.0  TV-MA    99 min     6.2/10        Movie  September 8, 2020
2        2019.0  TV-14    95 min     6.4/10        Movie       July 1, 2020
3        2020.0  TV-MA  1 Season     6.6/10      TV Show                NaN
4        2020.0  TV-14    90 min     5.1/10        Movie   February 5, 2020
```

في الانطباعات الأولى على مجموعة البيانات، أستطيع أن أرى أن عمود العنوان (Title column) يحتاج إلى إعداد لأنه يحتوي على # قبل اسم الأفلام أو البرامج التلفزيونية. سوف أعود إليها.في الوقت الحالي، دعنا نلقي نظرة على ما إذا كانت البيانات تحتوي على قيم خالية أم لا:

```
print(data.isnull().sum())
```

```
Show Id                0
Title                  0
Description            0
Director            2064
Genres                 0
Cast                 530
Production Country   559
Release Date           3
Rating                 4
Duration               3
Imdb Score           608
Content Type           0
Date Added          1335
dtype: int64
```

تحتوي مجموعة البيانات على قيم فارغة، ولكن قبل إزالة القيم الخالية، دعنا نحدد الأعمدة التي يمكننا استخدامها لبناء نظام توصية Netflix:

```
data = data[["Title", "Description", "Content Type",
"Genres"]]
print(data.head())
```

```
                          Title  \
0                      (Un)Well
1                        #Alive
2  #AnneFrank - Parallel Stories
3                      #blackAF
4              #cats_the_mewvie

                                         Description Content Type  \
0  This docuseries takes a deep dive into the luc...      TV Show
1  As a grisly virus rampages a city, a lone man ...        Movie
2  Through her diary, Anne Frank's story is retol...        Movie
3  Kenya Barris and his family navigate relations...      TV Show
4  This pawesome documentary explores how our fel...        Movie

                                          Genres
0                                     Reality TV
1  Horror Movies, International Movies, Thrillers
2             Documentaries, International Movies
3                                     TV Comedies
4             Documentaries, International Movies
```

حسب الاسم المقترح:

1. يحتوي عمود العنوان (title column) على عناوين الأفلام والبرامج التلفزيونية على Netflix.
2. يصف عمود الوصف (Description column) مخطط العروض التلفزيونية والأفلام.

3. يخبرنا عمود (Content Type column) ما إذا كان فيلمًا أم عرضًا تلفزيونيًا.
4. يحتوي عمود النوع (Genre column) على جميع أنواع العرض التلفزيوني أو الفيلم.

الآن دعنا نسقط الصفوف التي تحتوي على قيم خالية وننتقل إلى أبعد من ذلك:

```
data = data.dropna()
```

الآن سوف أقوم بتنظيف عمود العنوان لأنه يحتوي على بعض إعداد البيانات:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["Title"] = data["Title"].apply(clean)
```

دعنا الآن نلقي نظرة على بعض عينات العناوين قبل المضي قدمًا:

```
print(data.Title.sample(10))
```

```
3111          miniforc super dino power
1822                        girl reveng
910                       casino tycoon
4075                         sand castl
2760                               lock
3406                         nightflyer
536     bangkok love stori object affect
4365                            special
1733                               full
2343                     jeff dunham map
Name: Title, dtype: object
```

الآن سأستخدم عمود الأنواع كميزة للتوصية بمحتوى مشابه للمستخدم. سأستخدم مفهوم تشابه جيب التمام (cosine similarity) هنا (يُستخدم للعثور على أوجه التشابه في مستندين):

```
feature = data["Genres"].tolist()
tfidf = text.TfidfVectorizer(input=feature,
stop_words="english")
tfidf_matrix = tfidf.fit_transform(feature)
similarity = cosine_similarity(tfidf_matrix)
```

الآن سأقوم بتعيين عمود العنوان كفهرس حتى نتمكن من العثور على محتوى مشابه من خلال إعطاء عنوان الفيلم أو العرض التلفزيوني كمدخل:

```
indices = pd.Series(data.index,
        index=data['Title']).drop_duplicates()
```

إليك الآن كيفية كتابة دالة للتوصية بالأفلام والبرامج التلفزيونية على Netflix:

```
def netFlix_recommendation(title, similarity = similarity):
  index = indices[title]
  similarity_scores = list(enumerate(similarity[index]))
  similarity_scores = sorted(similarity_scores, key=lambda x:
x[1], reverse=True)
  similarity_scores = similarity_scores[0:10]
  movieindices = [i[0] for i in similarity_scores]
  return data['Title'].iloc[movieindices]

print(netFlix_recommendation("girlfriend"))
```

```
3                        blackaf
285                   washington
417               arrest develop
434     astronomi club sketch show
451    aunti donna big ol hous fun
656                      big mouth
752               bojack horseman
805                   brew brother
935                       champion
937                  chappell show
Name: Title, dtype: object
```

هذه هي الطريقة التي يمكنك بها إنشاء نظام توصيات Netflix باستخدام لغة برمجة بايثون.

الملخص

يتنبأ نظام التوصية الخاص بـ Netflix بكتالوج مخصص لك استنادًا إلى عوامل مثل محفوظات العرض الخاصة بك وسجل عرض المستخدمين الآخرين ذوي الأذواق والتفضيلات المماثلة والأنواع والفئة والأوصاف والمزيد من المعلومات عن المحتوى الذي شاهدته. آمل أن تكون قد أحببت هذه المقالة حول إنشاء نظام توصيات Netflix باستخدام بايثون.

10) التنبؤ بترافيك موقع ويب باستخدام بايثون Website Traffic Forecasting using Python

يعني التنبؤ بترافيك موقع الويب (Website Traffic Forecasting) التنبؤ بترافيك موقع الويب خلال فترة معينة. إنها واحدة من أفضل حالات استخدام تنبؤ السلاسل الزمنية (Time Series Forecasting). إذا كنت تريد معرفة كيفية التنبؤ بترافيك موقع ويب، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة التنبؤ بترافيك موقع ويب باستخدام بايثون.

التنبؤ بترافيك موقع ويب باستخدام بايثون

يتم جمع مجموعة البيانات التي أستخدمها للتنبؤ بترافيك موقع الويب من بيانات الترافيك اليومية لموقع thecleverprogrammer.com. يحتوي على بيانات حول بيانات الترافيك اليومية من يونيو 2021 إلى يونيو 2022. يمكنك تنزيل مجموعة البيانات من هنا. لنبدأ الآنفي مهمة التنبؤ بترافيك موقع الويب عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.api as sm

data = pd.read_csv("Thecleverprogrammer.csv")
print(data.head())
```

```
         Date  Views
0  01/06/2021   7831
1  02/06/2021   7798
2  03/06/2021   7401
3  04/06/2021   7054
4  05/06/2021   7973
```

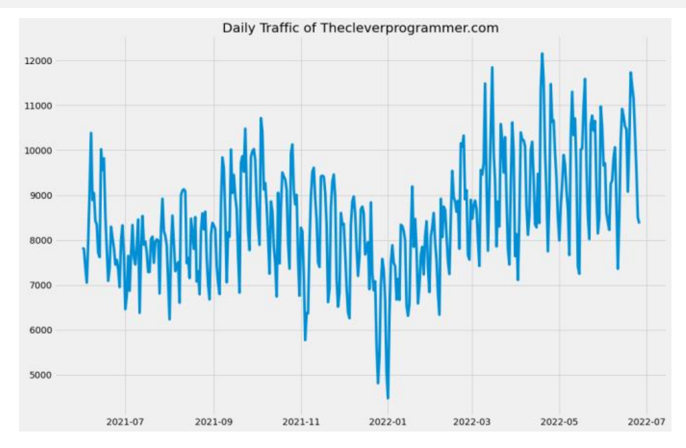
تحتوي مجموعة البيانات على عمودين، التاريخ (date) والترافيك(traffic). قبل المضي قدمًا، سأحول عمود التاريخ إلى نوع بيانات التاريخ والوقت:

```
data["Date"] = pd.to_datetime(data["Date”],
            format="%d/%m/%Y”)
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 391 entries, 0 to 390
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    391 non-null    datetime64[ns]
 1   Views   391 non-null    int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 6.2 KB
None
```

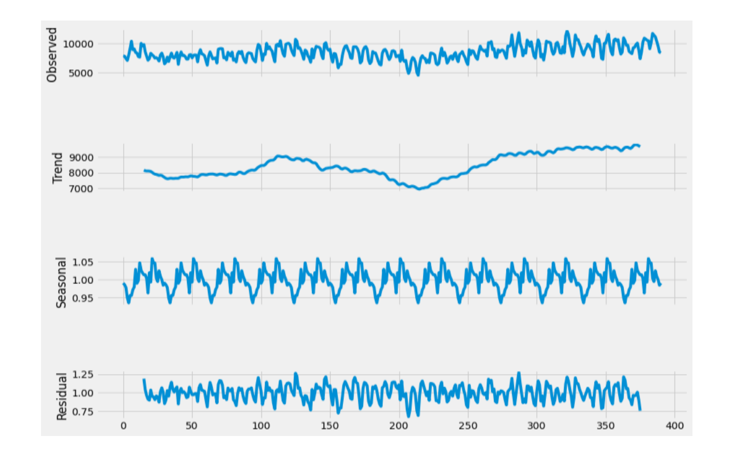
كان عمود التاريخ والوقت كائنًا (object)في البداية، لذلك قمت بتحويله إلى عمود التاريخ والوقت. دعنا الآن نلقي نظرة على الترافيك اليومية لموقع الويب:

```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(15, 10))
plt.plot(data["Date"], data["Views"])
plt.title("Daily Traffic of Thecleverprogrammer.com")
plt.show()
```



بيانات الترافيك على موقعنا موسمية (seasonal) لأن الترافيك على الموقع تزداد خلال أيام الأسبوع وتنخفض خلال عطلات نهاية الأسبوع. من المفيد معرفة ما إذا كانت مجموعة البيانات موسمية أم لا أثناء العمل على مشكلة تنبؤ السلاسل الزمنية. فيما يلي كيف يمكننا إلقاء نظرة على ما إذا كانت مجموعة البيانات الخاصة بنا ثابتة (stationary) أو موسمية(seasonal):

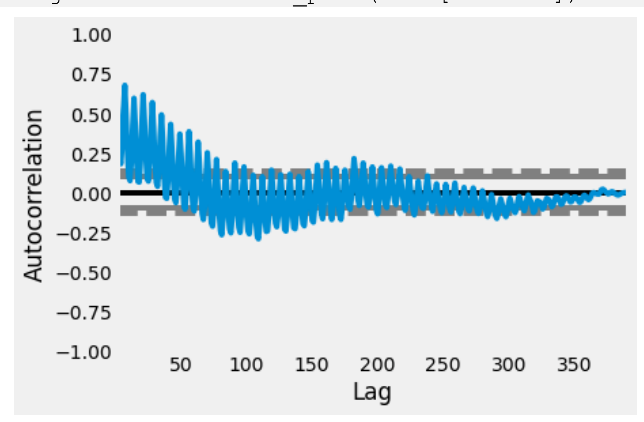
```
result = seasonal_decompose(data["Views"],
            model='multiplicative',
            freq = 30)
fig = plt.figure ()
fig = result.plot ()
fig.set_size_inches(10 ,15)
```



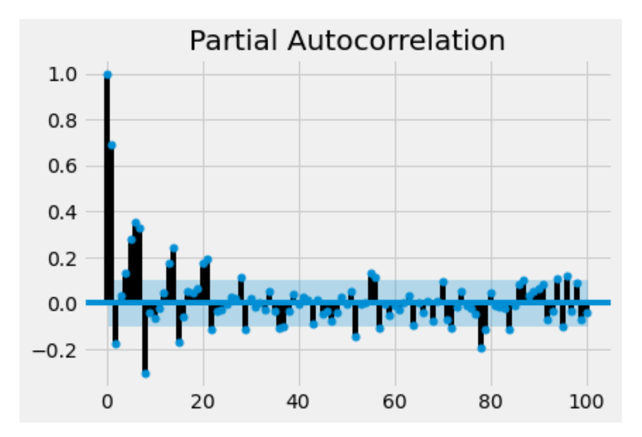
سأستخدم نموذج ARIMA الموسمي (SARIMA) للتنبؤ بالترافيك على موقع الويب. قبل استخدام نموذج SARIMA، من الضروري إيجاد قيم p و d و q. يمكنك معرفة كيفية العثور على قيم p و d و q من هنا.

نظرًا لأن البيانات ليست ثابتة، فإن قيمة d هي 1. للعثور على قيم p و q، يمكننا استخدام مخططات الارتباط التلقائي (autocorrelation) والارتباط التلقائي الجزئي (partial autocorrelation):

```
pd.plotting.autocorrelation_plot(data["Views"])
```




```
plot_pacf(data["Views"], lags = 100)
```



الآن إليك كيف يمكننا تدريب نموذج SARIMA لمهمة التنبؤ بترافيك موقع الويب:

p, d, q = 5, 1, 2

```
model=sm.tsa.statespace.SARIMAX(data['Views”],
                  order=(p, d, q),
                  seasonal_order=(p, d, q, 12))
model=model.fit()
print(model.summary())
```

```
                                    Statespace Model Results
==========================================================================================
Dep. Variable:                            Views   No. Observations:                    391
Model:             SARIMAX(5, 1, 2)x(5, 1, 2, 12)   Log Likelihood               -3099.402
Date:                          Tue, 28 Jun 2022   AIC                           6228.803
Time:                                  07:01:10   BIC                           6287.827
Sample:                                       0   HQIC                          6252.229
                                          - 391
Covariance Type:                            opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.7808      0.134      5.836      0.000       0.519       1.043
ar.L2         -0.7973      0.135     -5.920      0.000      -1.061      -0.533
ar.L3         -0.1442      0.170     -0.850      0.395      -0.477       0.188
ar.L4         -0.1833      0.151     -1.210      0.226      -0.480       0.114
ar.L5         -0.1548      0.139     -1.117      0.264      -0.426       0.117
ma.L1         -1.1826      0.094    -12.515      0.000      -1.368      -0.997
ma.L2          0.8856      0.078     11.304      0.000       0.732       1.039
ar.S.L12      -0.2606      4.608     -0.057      0.955      -9.293       8.772
ar.S.L24       0.0428      0.781      0.055      0.956      -1.488       1.573
ar.S.L36      -0.1880      0.246     -0.764      0.445      -0.670       0.294
ar.S.L48      -0.2151      0.959     -0.224      0.823      -2.095       1.664
ar.S.L60       0.0127      0.986      0.013      0.990      -1.920       1.946
ma.S.L12      -0.6902      4.611     -0.150      0.881      -9.728       8.348
ma.S.L24      -0.0994      3.637     -0.027      0.978      -7.228       7.029
sigma2      1.257e+06   1.59e+05      7.914      0.000    9.46e+05    1.57e+06
===================================================================================
Ljung-Box (Q):                      102.98   Jarque-Bera (JB):                  1.32
Prob(Q):                              0.00   Prob(JB):                          0.52
Heteroskedasticity (H):               1.03   Skew:                              0.14
Prob(H) (two-sided):                  0.85   Kurtosis:                          3.01
===================================================================================
```

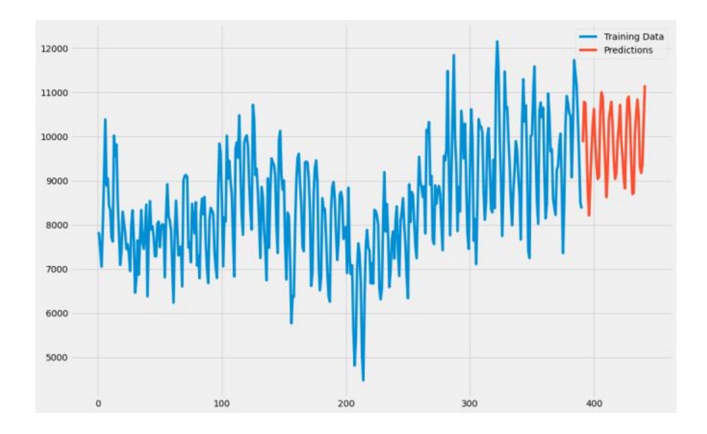
دعنا الآن نتنبأ بترافيك موقع الويب خلال الخمسين يومًا القادمة:

```
predictions = model.predict(len(data), len(data)+50)
print(predictions)
```

```
391     9874.390136
392    10786.957398
393    10757.445305
394     9863.890552
395     8765.031698
396     8212.310651
397     8929.181869
398     9685.809771
399    10270.622236
400    10625.904093
401     9854.870630
402     9362.193417
403     9040.021193
404     9081.558484
405    10538.993124
406    11003.816870
407    10897.859601
408    10083.291284
409     9445.806523
410     8629.901288
411     9184.420361
412    10392.770399
413    10593.941868
414    10788.128238
415    10263.101427
416     9449.467789
417     9040.226113
418     9168.972091
419     9887.094079
420    10218.658067
421    10715.657122
422     9899.224399
423     9541.622897
424     9065.810941
425     8825.335634
426    10137.936392
427    10839.866240
428    10905.862922
429    10411.640309
430     9451.211368
431     8698.339931
432     8725.534103
433    10060.678587
434    10506.263524
435    10842.515622
436    10485.387495
437     9335.244813
438     9175.122336
439     9357.034382
440    10295.910655
441    11162.934817
dtype: float64
```

إليك كيف يمكننا رسم التنبؤات:

```
data["Views"].plot(legend=True, label="Training Data”,
        figsize=(15, 10))
predictions.plot(legend=True, label="Predictions")
```



الملخص

إذن هذه هي الطريقة التي يمكنك بها التنبؤ بترافيك موقع الويب لفترة معينة. يعد التنبؤ بترافيك موقع الويب أحد أفضل أفكار مشروع علم البيانات التي يمكنك ذكرهافي سيرتك الذاتية. آمل أن يكون هذا المقال مفيداً لك لتعلم التنبؤ بترافيك موقع الويب باستخدام لغة برمجة بايثون.

11) نظام توصية المطاعم باستخدام بايثون Restaurant Recommendation System using Python

يعد نظام التوصية (recommendation system) أحد التطبيقات الشائعة لعلم البيانات. نظام توصية المطاعم (restaurant recommendation system) هو أحد التطبيقات التي توصي بمطاعم مماثلة للعميل وفقًا لذوق العميل. إذا كنت تريد معرفة كيفية إنشاء نظام توصية مطعم، فهذه المقالة مناسبة لك. ستأخذك هذه المقالةفي جولة حول كيفية إنشاء نظام توصية مطعم باستخدام بايثون.

نظام توصية المطاعم باستخدام بايثون

نظام توصية المطاعم هو أحد التطبيقات التي توصي بمطاعم مماثلة للعميل وفقًا لذوق العميل. لبناء نظام توصية مطعم باستخدام بايثون، قمت بجمع البيانات من Kaggle. يمكنك تنزيل مجموعة البيانات لهذه المهمة من هنا.

دعنا الآن نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import cosine_similarity

data =
pd.read_csv("TripAdvisor_RestauarantRecommendation.csv")
print(data.head())
```

```
                      Name       Street Address  \
0  Betty Lou's Seafood and Grill     318 Columbus Ave
1              Coach House Diner       55 State Rt 4
2               Table Talk Diner  2521 South Rd Ste C
3                   Sixty Vines     3701 Dallas Pkwy
4                  The Clam Bar    3914 Brewerton Rd

                    Location                                        Type  \
0  San Francisco, CA 94133-3908  Seafood, Vegetarian Friendly, Vegan Options
1     Hackensack, NJ 07601-6337         Diner, American, Vegetarian Friendly
2   Poughkeepsie, NY 12601-5476         American, Diner, Vegetarian Friendly
3          Plano, TX 75093-7777      American, Wine Bar, Vegetarian Friendly
4             Syracuse, NY 13212                       American, Bar, Seafood

           Reviews No of Reviews  \
0  4.5 of 5 bubbles   243 reviews
1    4 of 5 bubbles    84 reviews
2    4 of 5 bubbles   256 reviews
3  4.5 of 5 bubbles   235 reviews
4    4 of 5 bubbles   285 reviews
```

```
                                           Comments    Contact Number  \
0                                               NaN  +1 415-757-0569
1  Both times we were there very late, after 11 P...  +1 201-488-4999
2  Waitress was very friendly but a little pricey...  +1 845-849-2839
3  Not sure why I went there for the second time....  +1 469-620-8463
4  Doesn't look like much from the outside but wa...  +1 315-458-1662

                                    Trip_advisor Url  \
0  https://www.tripadvisor.com//Restaurant_Review...
1  https://www.tripadvisor.com//Restaurant_Review...
2  https://www.tripadvisor.com//Restaurant_Review...
3  https://www.tripadvisor.com//Restaurant_Review...
4  https://www.tripadvisor.com//Restaurant_Review...

                                        Menu Price_Range
0                  Check The Website for a Menu    $$ - $$$
1                  Check The Website for a Menu    $$ - $$$
2  http://tabletalkdiner.com/menu/breakfast/    $$ - $$$
3       https://sixtyvines.com/menu/plano-tx/    $$ - $$$
4                  Check The Website for a Menu    $$ - $$$
```

سأحدد عمودين من مجموعة البيانات لبقية المهمة (الاسم (Name) والنوع (Type)):

```
data = data[["Name", "Type"]]
print(data.head())
```

```
                          Name                                   Type
0  Betty Lou's Seafood and Grill  Seafood, Vegetarian Friendly, Vegan Options
1              Coach House Diner          Diner, American, Vegetarian Friendly
2               Table Talk Diner          American, Diner, Vegetarian Friendly
3                    Sixty Vines       American, Wine Bar, Vegetarian Friendly
4                   The Clam Bar                        American, Bar, Seafood
```

قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت البيانات تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
Name     0
Type    13
dtype: int64
```

لذلك تحتوي البيانات على بعض القيم الخاليةفي عمود النوع. سأحذف الصفوف التي تحتوي على قيم فارغة قبل المضي قدمًا:

```
data = data.dropna()
```

يعد نوع المطعم ميزة قيّمةفي البيانات لبناء نظام توصية. يمثل عمود النوع هنا فئة المطاعم. على سبيل المثال، إذا كان العميل يحب المطاعم الصديقة للنباتيين، فسوف ينظر فقط إلى التوصيات إذا كانت صديقة للنباتيين أيضًا. لذلك سأستخدم عمود النوع كميزة للتوصية بمطاعم مماثلة للعميل:

```
feature = data["Type"].tolist()
tfidf = text.TfidfVectorizer(input=feature,
stop_words="english")
tfidf_matrix = tfidf.fit_transform(feature)
similarity = cosine_similarity(tfidf_matrix)
```

الآن سأقوم بتعيين اسم المطعم كفهرس حتى نتمكن من العثور على مطاعم مماثلة من خلال إعطاء اسم المطعم كمدخل:

```
indices = pd.Series(data.index,
index=data['Name']).drop_duplicates()
```

الآن إليك كيفية كتابة دالة للتوصية بمطاعم مماثلة:

```
def restaurant_recommendation(name, similarity = similarity):
  index = indices[name]
  similarity_scores = list(enumerate(similarity[index]))
  similarity_scores = sorted(similarity_scores, key=lambda x:
x[1], reverse=True)
  similarity_scores = similarity_scores[0:10]
  restaurantindices = [i[0] for i in similarity_scores]
  return data['Name'].iloc[restaurantindices]

print(restaurant_recommendation("Market Grill"))
```

```
23                  The Lion's Share
154                        Houlihan's
518             Midgley's Public House
568                  Aspen Creek Grill
770                Pete's Sunset Grille
1190       Paul Martin's American Grill
1581                     Aviation Grill
1872                     Aviation Grill
2193                  Crest Bar & Grill
2612     Tahoe Joe's Famous Steakhouse
Name: Name, dtype: object
```

الملخص

هذه هي كيفية بناء نظام التوصية بالمطعم باستخدام لغة برمجة بايثون. نظام التوصية بالمطعم هو تطبيق يوصي بمطاعم مماثلة للعميل وفقًا لذوق العميل. آمل أن تكون قد أحببت هذه المقالة حول بناء نظام التوصية بالمطعم باستخدام بايثون.

12) تحليل أداء اللاعب Virat Kohli باستخدام بايثون
Virat Kohli Performance Analysis using Python

يعد تحليل أداء اللاعب (player's performance) إحدى حالات استخدام علوم البيانات في التحليلات الرياضية (sports analytics). يعتبر فيرات كوهلي (Virat Kohli) أحد أشهر لاعبي الكريكيت في العالم. لذلك سيكون مشروع علم بيانات رائعًا إذا قمنا بتحليل أداء الضرب لـ Virat Kohli على مر السنين. لذلك إذا كنت تريد معرفة كيفية تحليل أداء Virat Kohli، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة تحليل أداء Virat Kohli باستخدام بايثون.

تحليل أداء Virat Kohli (دراسة حالة)

يعتبر Virat Kohli أحد أشهر لاعبي الكريكيت في العالم. هنا تحصل على مجموعة بيانات لجميع مباريات ODI التي لعبها Virat Kohli من 18 أغسطس 2008 إلى 22 يناير 2017. أنت مطالب بتحليل أداء Virat Kohli في مباريات ODI.

فيما يلي المعلومات الكاملة حول جميع الأعمدة في مجموعة البيانات:

1. **Runs**: النقاط في المباراة.
2. **BF**: الكرات التي تواجهها في المباراة.
3. **4s**: عدد 4s في المباراة.
4. **6s**: عدد 6s في المباراة.
5. **SR**: معدل الضربات في المباراة.
6. **Pos**: مركز الضرب في المباراة.
7. **Dismissal**: كيف خرج Virat Kohli في المباراة.
8. **Inns**: الشوط الأول والثاني.
9. **Opposition**: من كان خصم الهند.
10. **Ground**: مكان المباراة.
11. **Start Date**: تاريخ المباراة.

يمكنك تنزيل مجموعة البيانات هذه من هنا.

تحليل أداء Virat Kohli باستخدام بايثون

لنبدأ الآن بمهمة تحليل أداء Virat Kohli باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("Virat_Kohli.csv")
print(data.head())
```

```
   Runs  BF  4s  6s     SR  Pos Dismissal  Inns   Opposition         Ground  \
0    12  22   1   0  54.54  2.0       lbw     1  v Sri Lanka       Dambulla
1    37  67   6   0  55.22  2.0    caught     2  v Sri Lanka       Dambulla
2    25  38   4   0  65.78  1.0   run out     1  v Sri Lanka  Colombo (RPS)
3    54  66   7   0  81.81  1.0    bowled     1  v Sri Lanka  Colombo (RPS)
4    31  46   3   1  67.39  1.0       lbw     2  v Sri Lanka  Colombo (RPS)

  Start Date
0  18-Aug-08
1  20-Aug-08
2  24-Aug-08
3  27-Aug-08
4  29-Aug-08
```

دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا قبل المضي قدمًا:

```
print(data.isnull().sum())
```

```
Runs          0
BF            0
4s            0
6s            0
SR            0
Pos           0
Dismissal     0
Inns          0
Opposition    0
Ground        0
Start Date    0
dtype: int64
```

تحتوي مجموعة البيانات على المباريات التي لعبها Virat Kohli بين 18 أغسطس 2008 و 22 يناير 2017. لذلك دعونا نلقي نظرة على إجمالي النقاط التي سجلها Virat Kohli:

```
#Total Runs Between 18-Aug-08 - 22-Jan-17
data["Runs"].sum()
```

```
6184
```

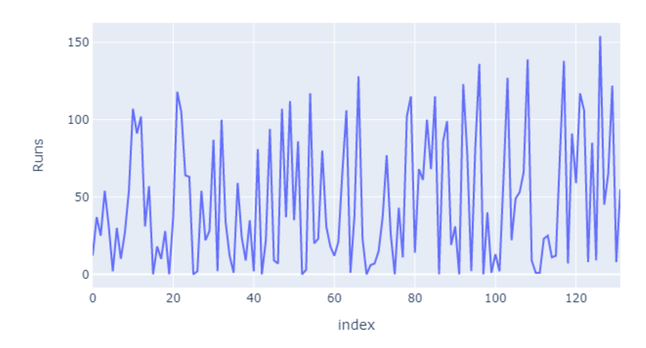
الآن دعونا نلقي نظرة على متوسط نقاط Virat Kohli خلال نفس الفترة:

```
#Average Runs Between 18-Aug-08 - 22-Jan-17
data["Runs"].mean()
```

```
46.84848484848485
```

في ODIs، يعتبر متوسط الضرب من 35ـ37 متوسطاً جيداً. لذا فإن معدل الضرب في Virat Kohli جيد. الآن دعونا نلقي نظرة على اتجاه الجري الذي سجله Virat Kohli في مسيرته من 18 أغسطس 2008 إلى 22 يناير 2017:

```
matches = data.index
figure = px.line(data, x=matches, y="Runs",
       title='Runs Scored by Virat Kohli Between 18-Aug-08 -
22-Jan-17')
figure.show()
```



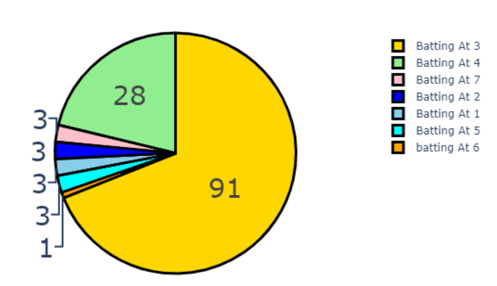
في العديد من الاشواط التي لعبها Virat Kohli ، سجل أكثر من 100 نقطة أو ما يقرب من ذلك. هذه علامة جيدة على الثبات. الآن دعونا نرى جميع مواقع الضرب (batting positions) التي لعبها Virat Kohli :

```
#Batting Positions
data["Pos"] = data["Pos"].map({3.0: "Batting At 3", 4.0:
"Batting At 4", 2.0: "Batting At 2,"
":1.0             Batting At 1", 7.0:"Batting At 7",
5.0:"Batting At 5,"
":6.0             batting At 6"})

Pos = data["Pos"].value_counts()
label = Pos.index
counts = Pos.values
```

```
colors = ['gold','lightgreen', "pink", "blue", "skyblue",
"cyan", "orange"]

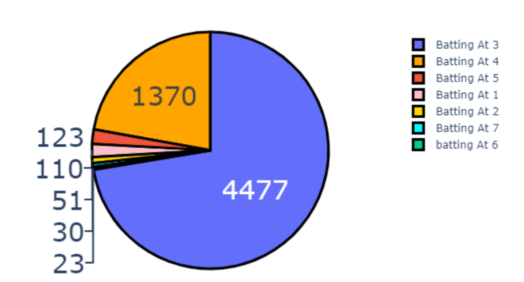
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Number of Matches At Different
Batting Positions')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3))(
fig.show()
```



في أكثر من 68٪ من جميع الاشواط التي لعبها Virat Kohli ، احتل المركز الثالث. الآن دعونا نلقي نظرة على مجموع النقاط (total runs) التي سجلها Virat Kohli في مراكز مختلفة:

```
label = data["Pos"]
counts = data["Runs"]
colors = ['gold','lightgreen', "pink", "blue", "skyblue",
"cyan", "orange"]

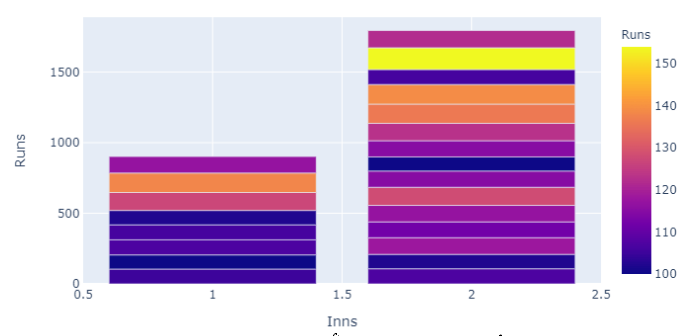
fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Runs By Virat Kohli At Different
Batting Positions')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



أكثر من 72٪ من مجموع النقاط التي سجلها Virat Kohli هي في المركز الثالث. لذلك يمكننا القول أن الضرب في المركز الثالث مثالي لـVirat Kohli .

الآن دعونا نلقي نظرة على عدد القرون centuries (100 نقطة في شوط واحد) التي سجلها Virat Kohli أثناء الضرب في الاشواط الأولى والثانية:

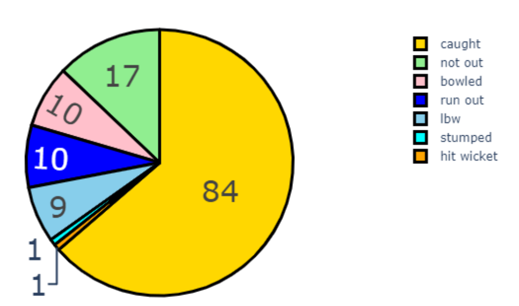
```
centuries = data.query("Runs >= 100")
figure = px.bar(centuries, x=centuries["Inns"], y =
centuries["Runs"] ,
       color = centuries["Runs"],
       title="Centuries By Virat Kohli in First Innings Vs.
Second Innings")
figure.show()
```



لذلك يتم تسجيل معظم القرون أثناء الضرب في الأدوار الثانية. من خلال هذا، يمكننا القول إن Virat Kohli يحب مطاردة النتائج. دعنا الآن نلقي نظرة على نوع عمليات الطرد (dismissals) التي واجهها Virat Kohli في معظم الأوقات:

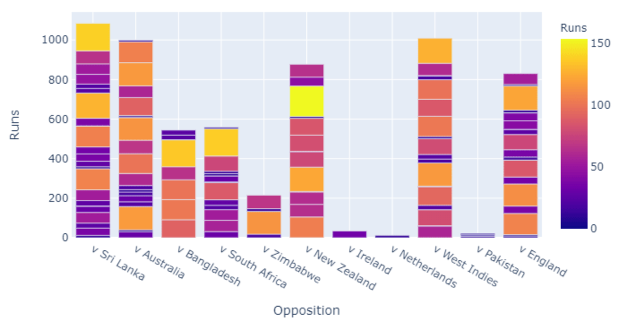
```
#Dismissals of Virat Kohli
dismissal = data["Dismissal"].value_counts()
label = dismissal.index
counts = dismissal.values
colors = ['gold','lightgreen', "pink", "blue", "skyblue",
"cyan", "orange"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Dismissals of Virat Kohli')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



لذلكفي معظم الأوقات، يخرج Virat Kohli من خلال القبض عليه من قبل اللاعب أو الحارس. الآن دعونا نلقي نظرة على الفريق الذي سجل Virat Kohli معظم نقاطه:

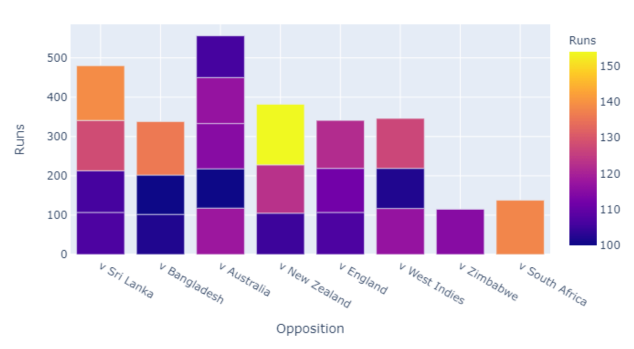
```
figure = px.bar(data, x=data["Opposition"], y = data["Runs"],
color = data["Runs”],
     title="Most Runs Against Teams”)
figure.show()
```



وفقًا للرقم أعلاه، يحب Virat Kohli اللعب ضد سريلانكا وأستراليا ونيوزيلندا وجزر الهند الغربية وإنجلترا. لكنه سجل معظم أشواطه أثناء لعبه ضد سريلانكا. دعونا الآن نلقي نظرة على الفريق الذي سجل ضده Virat Kohli معظم قرونه:

```
figure = px.bar(centuries, x=centuries["Opposition"], y =
centuries["Runs”],
       color = centuries["Runs"],
       title="Most Centuries Against Teams”)
figure.show()
```





لذلك، كانت معظم القرون التي سجلها Virat Kohli ضد أستراليا. الآن دعونا نحلل معدل ضربات Virat Kohli. لتحليل معدل ضربات Virat Kohli ، سأقوم بإنشاء مجموعة بيانات جديدة لجميع المباريات التي لعبها Virat Kohli حيث كان معدل تسديده أكثر من 120:

```
strike_rate = data.query("SR >= 120")
print(strike_rate)
```

```
     Runs  BF  4s  6s      SR           Pos Dismissal  Inns      Opposition  \
8      27  19   4   0  142.10  Batting At 7    bowled     1     v Sri Lanka
32    100  83   8   2  120.48  Batting At 4   not out     1    v Bangladesh
56     23  11   3   0  209.09  batting At 6   not out     1  v West Indies
76     43  34   4   1  126.47  Batting At 3    caught     1       v England
78    102  83  13   2  122.89  Batting At 3    caught     1  v West Indies
83    100  52   8   7  192.30  Batting At 3   not out     2     v Australia
85    115  66  18   1  174.24  Batting At 3   not out     2     v Australia
93     78  65   7   2  120.00  Batting At 3    caught     2  v New Zealand
130     8   5   2   0  160.00  Batting At 3    caught     1       v England

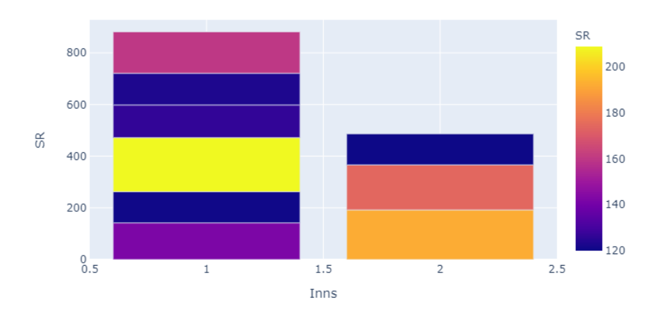
             Ground Start Date
8            Rajkot  15-Dec-09
32            Dhaka  19-Feb-11
56           Indore   8-Dec-11
76       Birmingham  23-Jun-13
78    Port of Spain   5-Jul-13
83           Jaipur  16-Oct-13
85           Nagpur  30-Oct-13
93         Hamilton  22-Jan-14
130         Cuttack  19-Jan-17
```

الآن دعونا نرى ما إذا كان Virat Kohli يلعب بمعدلات ضربات عالية في الاشواط الأولى أو الاشواط الثانية:

```
figure = px.bar(strike_rate, x = strike_rate["Inns”],
        y = strike_rate["SR"],
        color = strike_rate["SR"],
      title="Virat Kohli's High Strike Rates in First Innings
Vs. Second Innings”)
```

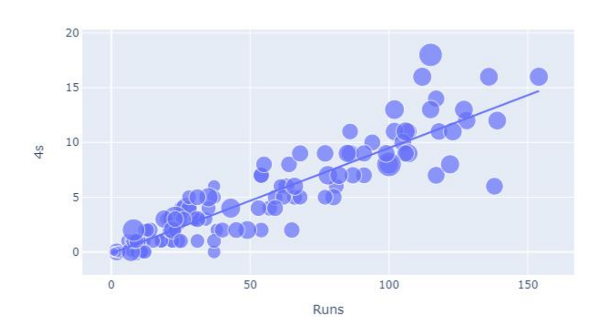
```
figure.show()
```





لذلك وفقًا للشكل أعلاه، يحب Virat Kohli اللعب بقوة أكبرفي الاشواط الأولى مقارنة بالاشواط الثانية. الآن دعونا نرى العلاقة بين الأشواط التي سجلها Virat Kohli والأربع (fours) التي لعبهافي كل شوط:

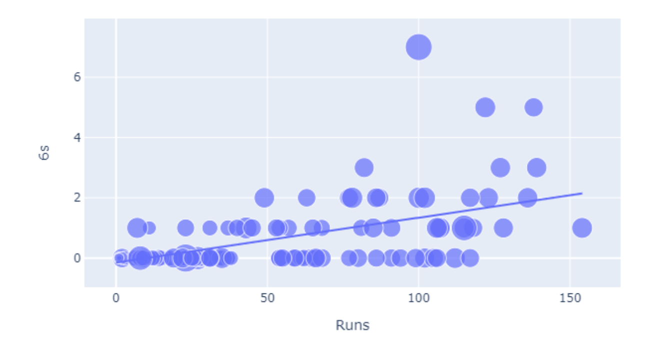
```
figure = px.scatter(data_frame = data, x="Runs,"
        y="4s", size="SR", trendline="ols,"
        title="Relationship Between Runs Scored and Fours")
figure.show()
```



هناك علاقة خطية. هذا يعني أن Virat Kohli يحب اللعب الرباعي. كلما زاد عدد مرات الجري التي سجلهافي الأدوار، كلما لعب رباعيات أكثر. دعونا نرى ما إذا كانت هناك علاقة ما مع السداسيات (sixes):

```
figure = px.scatter(data_frame = data, x="Runs,"
        y="6s", size="SR", trendline="ols,"
        title= "Relationship Between Runs Scored and Sixes")
figure.show()
```





لا توجد علاقة خطية قوية هنا. هذا يعني أن Virat Kohli يحب لعب رباعيات أكثر من سداسيات. هذه هي الطريقة التي يمكنك بها تحليل أداء Virat Kohli أو أي لاعب كريكيت آخرفي العالم.

الملخص

هذه هي الطريقة التي يمكنك بها إجراء تحليل أداء Virat Kohli باستخدام لغة برمجة بايثون. يعد تحليل أداء اللاعب إحدى حالات استخدام علم البياناتفي التحليلات الرياضية. آمل أن تكون قد أحببت هذه المقالة حول تحليل أداء Virat Kohli باستخدام بايثون.

13) نظام توصية الكتب باستخدام بايثون Book Recommendation System using Python

يعد نظام التوصية (recommendation system) أحد التطبيقات الشائعة لعلم البيانات. نظام توصية الكتب (Book Recommendation system) هو تطبيق يستخدم للتوصية بالكتب المماثلة للمستخدم. إذا كنت تريد معرفة كيفية إنشاء نظام توصية للكتب، فهذه المقالة مناسبة لك. ستأخذك هذه المقالةفي جولة حول كيفية إنشاء نظام توصية للكتب باستخدام بايثون.

نظام توصية الكتب باستخدام بايثون

يجب أن يوصي نظام التوصية الكتب بالكتب المماثلة بناءً على اهتمام المستخدم. يتم جمع مجموعة البيانات اللازمة لبناء نظام توصية الكتاب من Kaggle. يمكنك تنزيل مجموعة البيانات من هنا.

دعنا الآن نستورد مكتبات بايثون الضرورية ومجموعة البيانات التي نحتاجها لهذه المهمة:

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.metrics.pairwise import linear_kernel

data = pd.read_csv("book_data.csv")
print(data.head())
```

```
                                         book_authors  ...                                          image_url
0                                     Suzanne Collins  ...  https://images.gr-assets.com/books/1447303603l...
1                          J.K. Rowling|Mary GrandPré  ...  https://images.gr-assets.com/books/1255614970l...
2                                          Harper Lee  ...  https://images.gr-assets.com/books/1361975680l...
3  Jane Austen|Anna Quindlen|Mrs. Oliphant|George...  ...  https://images.gr-assets.com/books/1320399351l...
4                                     Stephenie Meyer  ...  https://images.gr-assets.com/books/1361039443l...

[5 rows x 12 columns]
```

سأختار ثلاثة أعمدة من مجموعة البيانات لبقية المهمة (book_title، book_desc، book_rating_count):

```
data = data[["book_title", "book_desc", "book_rating_count"]]
print(data.head())
```

```
                                   book_title  ... book_rating_count
0                            The Hunger Games  ...           5519135
1  Harry Potter and the Order of the Phoenix  ...           2041594
2                       To Kill a Mockingbird  ...           3745197
3                         Pride and Prejudice  ...           2453620
4                                    Twilight  ...           4281268

[5 rows x 3 columns]
```

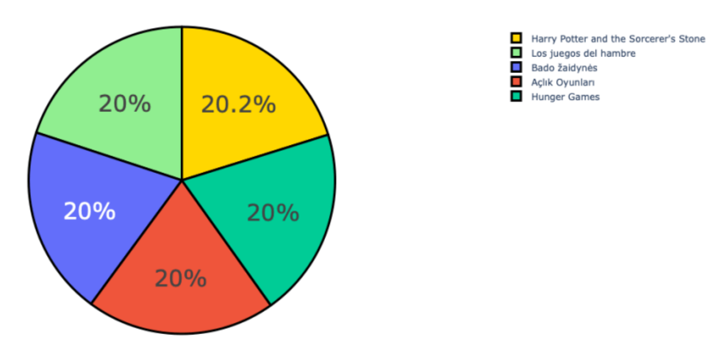
دعنا نلقي نظرة على أفضل 5 كتب في مجموعة البيانات وفقاً لعدد التقييمات:

```
data = data.sort_values(by="book_rating_count",
ascending=False)
top_5 = data.head()

import plotly.express as px
import plotly.graph_objects as go

labels = top_5["book_title"]
values = top_5["book_rating_count"]
colors = ['gold','lightgreen']


fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
fig.update_layout(title_text="Top 5 Rated Books")
fig.update_traces(hoverinfo='label+percent',
textinfo='percent', textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



قبل المضي قدماً، دعنا نتحقق مما إذا كانت البيانات تحتوي على قيم خالية أم لا:

```
print(data.isnull().sum())
```

```
book_title              0
book_desc            1331
book_rating_count       0
dtype: int64
```

تحتوي مجموعة البيانات على قيم خاليةفي عمود وصف الكتاب (book description column). دعنا نسقط الصفوف التي تحتوي على قيم خالية:

```
data = data.dropna()
```

سأستخدم الآن عمود وصف الكتاب كميزة للتوصية بالكتب المماثلة للمستخدم:

```
feature = data["book_desc"].tolist()
tfidf = text.TfidfVectorizer(input=feature,
stop_words="english")
tfidf_matrix = tfidf.fit_transform(feature)
similarity = linear_kernel(tfidf_matrix, tfidf_matrix)
```

الآن سأقوم بتعيين عمود عنوان الكتاب (book title column) كفهرس حتى نتمكن من العثور على كتب مماثلة من خلال إعطاء عنوان الكتاب كمدخل:

```
indices = pd.Series(data.index,
        index=data['book_title']).drop_duplicates()
```

الآن إليك كيفية كتابة دالة للتوصية بكتب مماثلة:

```
def book_recommendation(title, similarity = similarity):
  index = indices[title]
  similarity_scores = list(enumerate(similarity[index]))
  similarity_scores = sorted(similarity_scores, key=lambda x:
x[1], reverse=True)
  similarity_scores = similarity_scores[0:5]
  bookindices = [i[0] for i in similarity_scores]
  return data['book_title'].iloc[bookindices]

print(book_recommendation("Letters to a Secret Lover"))
```

```
21823    The Kabbalah of Jesus Christ, Part 1 The True ...
28960                    Seeing and Savoring Jesus Christ
17173                         Jesus and Moses in India
7944                            The Jesus I Never Knew
16976    Beautiful Outlaw: Experiencing the Playful, Di...
Name: book_title, dtype: object
```

الملخص

يجب أن يوصي نظام التوصية بالكتب المماثلة وفقًا لمصلحة المستخدم. كمبتدئ في علم البيانات، يجب أن تعمل بشكل جيد في مشروع علم البيانات هذا للتعرف على أنظمة التوصية. يمكنك العثور على المزيد من مشاريع علوم البيانات للتدرب عليها من هنا. أتمنى أن تكون قد أحببت هذه المقالة حول كيفية بناء نظام توصية كتاب باستخدام بايثون.

14) تحليل بيانات الساعات الذكية باستخدام بايثون

Smartwatch Data Analysis using Python

هناك الكثير من المنافسة بين العلامات التجاريةفي صناعة الساعات الذكية (smartwatch). الساعات الذكية مفضلة من قبل الأشخاص الذين يرغبونفي رعاية لياقتهم البدنية. يعد تحليل البيانات التي تم جمعها عن لياقتك إحدى حالات استخدام علم البياناتفي الرعاية الصحية. لذلك إذا كنت تريد معرفة كيفية تحليل بيانات اللياقة للساعة الذكية، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل بيانات الساعة الذكية باستخدام بايثون.

تحليل بيانات الساعة الذكية باستخدام بايثون

مجموعة البيانات التي أستخدمها لتحليل بيانات الساعة الذكية متاحة للجمهور على Kaggle. تم جمع مجموعة البيانات هذهفي البداية من 30 من مستخدمي ساعة Fitbit الذكية. يمكنك تنزيل مجموعة البيانات من هنا.

سأبدأ الآن مهمة تحليل بيانات الساعة الذكية عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("dailyActivity_merged.csv")
print(data.head())
```

```
          Id ActivityDate  TotalSteps  TotalDistance  TrackerDistance  \
0  1503960366    4/12/2016       13162           8.50             8.50
1  1503960366    4/13/2016       10735           6.97             6.97
2  1503960366    4/14/2016       10460           6.74             6.74
3  1503960366    4/15/2016        9762           6.28             6.28
4  1503960366    4/16/2016       12669           8.16             8.16

   LoggedActivitiesDistance  VeryActiveDistance  ModeratelyActiveDistance  \
0                       0.0                1.88                      0.55
1                       0.0                1.57                      0.69
2                       0.0                2.44                      0.40
3                       0.0                2.14                      1.26
4                       0.0                2.71                      0.41

   LightActiveDistance  SedentaryActiveDistance  VeryActiveMinutes  \
0                 6.06                      0.0                 25
1                 4.71                      0.0                 21
2                 3.91                      0.0                 30
3                 2.83                      0.0                 29
4                 5.04                      0.0                 36

   FairlyActiveMinutes  LightlyActiveMinutes  SedentaryMinutes  Calories
0                   13                   328               728      1985
1                   19                   217               776      1797
2                   11                   181              1218      1776
3                   34                   209               726      1745
4                   10                   221               773      1863
```

قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
Id                          0
ActivityDate                0
TotalSteps                  0
TotalDistance               0
TrackerDistance             0
LoggedActivitiesDistance    0
VeryActiveDistance          0
ModeratelyActiveDistance    0
LightActiveDistance         0
SedentaryActiveDistance     0
VeryActiveMinutes           0
FairlyActiveMinutes         0
LightlyActiveMinutes        0
SedentaryMinutes            0
Calories                    0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات على أي قيم فارغة. دعنا نلقي نظرة على المعلومات المتعلقة بالأعمدةفي مجموعة البيانات:

```
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Id                        940 non-null    int64
 1   ActivityDate              940 non-null    object
 2   TotalSteps                940 non-null    int64
 3   TotalDistance             940 non-null    float64
 4   TrackerDistance           940 non-null    float64
 5   LoggedActivitiesDistance  940 non-null    float64
 6   VeryActiveDistance        940 non-null    float64
 7   ModeratelyActiveDistance  940 non-null    float64
 8   LightActiveDistance       940 non-null    float64
 9   SedentaryActiveDistance   940 non-null    float64
 10  VeryActiveMinutes         940 non-null    int64
 11  FairlyActiveMinutes       940 non-null    int64
 12  LightlyActiveMinutes      940 non-null    int64
 13  SedentaryMinutes          940 non-null    int64
 14  Calories                  940 non-null    int64
dtypes: float64(7), int64(7), object(1)
memory usage: 110.3+ KB
None
```

العمود الذي يحتوي على تاريخ السجل هو كائن (object). قد نحتاج إلى استخدام التواريخفي تحليلنا، فلنحول هذا العمود إلى عمود التاريخ والوقت (datetime column):

```
#Changing datatype of ActivityDate
data["ActivityDate"] = pd.to_datetime(data["ActivityDate”],
                        format="%m/%d/%Y”)
print(data.info())
```

انظر إلى جميع الأعمدة؛ سترى معلومات حول الدقائق النشطة جداً، والنشطة إلى حدٍ ما، والنشطة الخفيفة، والدقائق المستقرة في مجموعة البيانات. دعنا نجمع كل هذه الأعمدة في إجمالي الدقائق قبل المضي قدمًا:

```
data["TotalMinutes"] = data["VeryActiveMinutes"] +
data["FairlyActiveMinutes"] + data["LightlyActiveMinutes"] +
data["SedentaryMinutes"]
print(data["TotalMinutes"].sample(5))
```

دعنا الآن نلقي نظرة على الإحصائيات الوصفية لمجموعة البيانات:

```
print(data.describe())
```

	Id	TotalSteps	TotalDistance	TrackerDistance
count	9.400000e+02	940.000000	940.000000	940.000000
mean	4.855407e+09	7637.910638	5.489702	5.475351
std	2.424805e+09	5087.150742	3.924606	3.907276
min	1.503960e+09	0.000000	0.000000	0.000000
25%	2.320127e+09	3789.750000	2.620000	2.620000
50%	4.445115e+09	7405.500000	5.245000	5.245000
75%	6.962181e+09	10727.000000	7.712500	7.710000
max	8.877689e+09	36019.000000	28.030001	28.030001

	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance
count	940.000000	940.000000	940.000000
mean	0.108171	1.502681	0.567543
std	0.619897	2.658941	0.883580
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.210000	0.240000
75%	0.000000	2.052500	0.800000
max	4.942142	21.920000	6.480000

	LightActiveDistance	SedentaryActiveDistance	VeryActiveMinutes
count	940.000000	940.000000	940.000000
mean	3.340819	0.001606	21.164894
std	2.040655	0.007346	32.844803
min	0.000000	0.000000	0.000000
25%	1.945000	0.000000	0.000000
50%	3.365000	0.000000	4.000000
75%	4.782500	0.000000	32.000000
max	10.710000	0.110000	210.000000

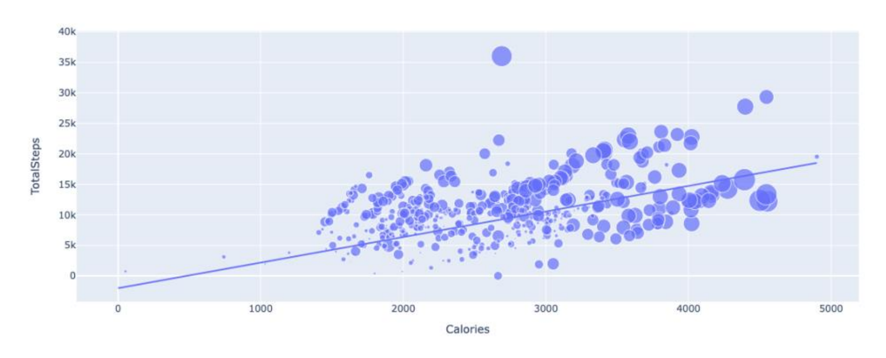
	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes
count	940.000000	940.000000	940.000000
mean	13.564894	192.812766	991.210638
std	19.987404	109.174700	301.267437
min	0.000000	0.000000	0.000000
25%	0.000000	127.000000	729.750000
50%	6.000000	199.000000	1057.500000
75%	19.000000	264.000000	1229.500000
max	143.000000	518.000000	1440.000000

	Calories	TotalMinutes
count	940.000000	940.000000
mean	2303.609574	1218.753191
std	718.166862	265.931767
min	0.000000	2.000000
25%	1828.500000	989.750000
50%	2134.000000	1440.000000
75%	2793.250000	1440.000000
max	4900.000000	1440.000000

تحليل بيانات الساعة الذكية

تحتوي مجموعة البيانات على عمود السعرات الحرارية (Calories)؛ يحتوي على بيانات حول عدد السعرات الحرارية المحروقةفي اليوم. دعونا نلقي نظرة على العلاقة بين حرق السعرات الحرارية (calories burned) وإجمالي الخطوات (total steps) التي تم قطعهافي اليوم:

```
figure = px.scatter(data_frame = data, x="Calories,"
        y="TotalSteps", size="VeryActiveMinutes,"
        trendline="ols,"
        title="Relationship between Calories & Total Steps")
figure.show()
```

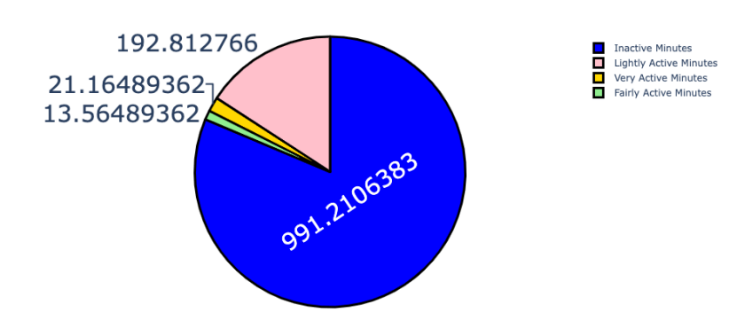


يمكنك أن ترى أن هناك علاقة خطية بين العدد الإجمالي للخطوات وعدد السعرات الحرارية المحروقةفي اليوم. دعونا الآن نلقي نظرة على متوسط العدد الإجمالي للدقائق النشطة (active minutes)في اليوم:

```
label = ["Very Active Minutes", "Fairly Active Minutes ",
"    Lightly Active Minutes", "Inactive Minutes "]
counts = data[["VeryActiveMinutes", "FairlyActiveMinutes",
"       LightlyActiveMinutes", "SedentaryMinutes"]].mean()
colors = ['gold','lightgreen', "pink", "blue"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Total Active Minutes')
```

```
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
         marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```



ملاحظات:

1. 81.3٪ من إجمالي الدقائق غير النشطة (inactive)في اليوم.
2. 15.8٪ من الدقائق النشطة بشكل خفيف (Lightly active)في اليوم.
3. في المتوسط، كانت 21 دقيقة فقط (1.74٪) نشطة للغاية (very active).
4. و1.11٪ (13 دقيقة) من الدقائق النشطة إلى حد ما (fairly active)في اليوم.

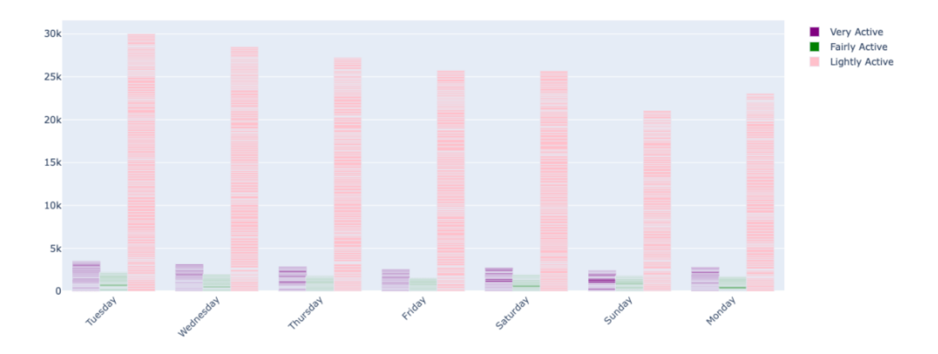
قمنا بتحويل نوع بيانات عمود (ActivityDate) إلى عمود التاريخ والوقت أعلاه. دعونا نستخدمه للعثور على أيام الأسبوع للسجلات وإضافة عمود جديد لمجموعة البيانات هذه باسم (Day):

```
data["Day"] = data["ActivityDate"].dt.day_name()
print(data["Day"].head())
```

دعنا الآن نلقي نظرة على الدقائق النشطة جداً، والنشطة إلى حدٍ ما، والنشطة الخفيفةفي كل يوم من أيام الأسبوع:

```
fig = go.Figure()
fig.add_trace(go.Bar)
  x=data["Day"],
  y=data["VeryActiveMinutes"],
  name='Very Active',
  marker_color='purple'
))
fig.add_trace(go.Bar)
```

```
  x=data["Day"],
  y=data["FairlyActiveMinutes"],
  name='Fairly Active',
  marker_color='green'
))
fig.add_trace(go.Bar)
  x=data["Day"],
  y=data["LightlyActiveMinutes"],
  name='Lightly Active',
  marker_color='pink,
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```

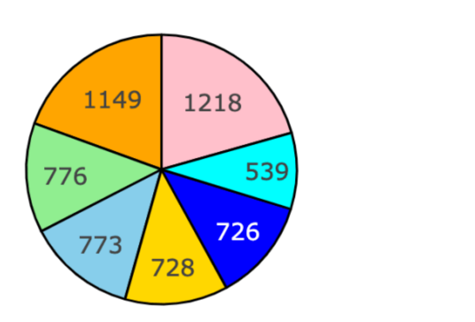


دعنا الآن نلقي نظرة على عدد الدقائق غير النشطةفي كل يوم من أيام الأسبوع:

```
day = data["Day"].value_counts()
label = day.index
counts = data["SedentaryMinutes"]
colors = ['gold','lightgreen', "pink", "blue", "skyblue",
"cyan", "orange"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Inactive Minutes Daily')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3))
fig.show()
```

Inactive Minutes Daily

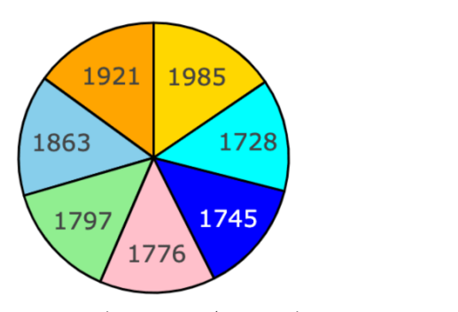


لذا فإن الخميس هو أكثر يوم غير نشط وفقًا لنمط حياة جميع الأفرادفي مجموعة البيانات. دعنا الآن نلقي نظرة على عدد السعرات الحرارية المحروقةفي كل يوم من أيام الأسبوع:

```
calories = data["Day"].value_counts()
label = calories.index
counts = data["Calories"]
colors = ['gold','lightgreen', "pink", "blue", "skyblue",
"cyan", "orange"]

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Calories Burned Daily')
fig.update_traces(hoverinfo='label+percent', textinfo='value',
textfont_size=30,
        marker=dict(colors=colors, line=dict(color='black',
width=3)))
fig.show()
```

Calories Burned Daily



لذلك، يُعد يوم الثلاثاء أحد أكثر الأيام نشاطًا لجميع الأفرادفي مجموعة البيانات، حيث تم حرق أكبر عدد من السعرات الحرارية يوم الثلاثاء.

هذه هي الطريقة التي يمكنك بها تحليل بيانات الساعة الذكية باستخدام لغة برمجة بايثون. هناك الكثير الذي يمكنك القيام به باستخدام مجموعة البيانات هذه. يمكنك أيضًا استخدامه للتنبؤ بعدد السعرات الحرارية المحروقةفي اليوم.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل البيانات التي جمعتها ساعة ذكية حول اللياقة البدنية باستخدام بايثون. الساعات الذكية مفضلة من قبل الأشخاص الذين يرغبونفي رعاية لياقتهم البدنية. يعد تحليل البيانات التي تم جمعها عن لياقتك إحدى حالات استخدام علم البياناتفي الرعاية الصحية. آمل أن تكون قد أحببت هذه المقالة حول تحليل بيانات الساعة الذكية باستخدام بايثون.

15) تحليل IPL 2022 باستخدام بايثون IPL 2022 Analysis using Python

ينتج عن كل حدث رياضي اليوم الكثير من البيانات حول اللعبة، والتي تُستخدم لتحليل أداء اللاعبين والفرق وكل حدث من أحداث اللعبة. لذا فإن استخدام علم البيانات موجودفي كل رياضة اليوم. حاليًا، يعد IPL 2022 أحد الأحداث الرياضية الشهيرة التي تقامفي الهند. لذلك، إذا كنت تريد معرفة كيفية تحليل IPL 2022، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل IPL 2022 باستخدام بايثون.

تحليل IPL 2022 باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل IPL 2022 من Kaggle. يمكنك تنزيل مجموعة البيانات هذه من هنا. لنبدأ الآن هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("IPL 2022.csv")
print(data.head())
```

```
   match_id            date                                   venue  \
0         1  March 26,2022               Wankhede Stadium, Mumbai
1         2  March 27,2022              Brabourne Stadium, Mumbai
2         3  March 27,2022     Dr DY Patil Sports Academy, Mumbai
3         4  March 28,2022               Wankhede Stadium, Mumbai
4         5  March 29,2022  Maharashtra Cricket Association Stadium,Pune

       team1      team2  stage toss_winner toss_decision  first_ings_score  \
0    Chennai    Kolkata  Group     Kolkata         Field               131
1      Delhi     Mumbai  Group       Delhi         Field               177
2   Banglore     Punjab  Group      Punjab         Field               205
3    Gujarat    Lucknow  Group     Gujarat         Field               158
4  Hyderabad  Rajasthan  Group   Hyderabad         Field               210

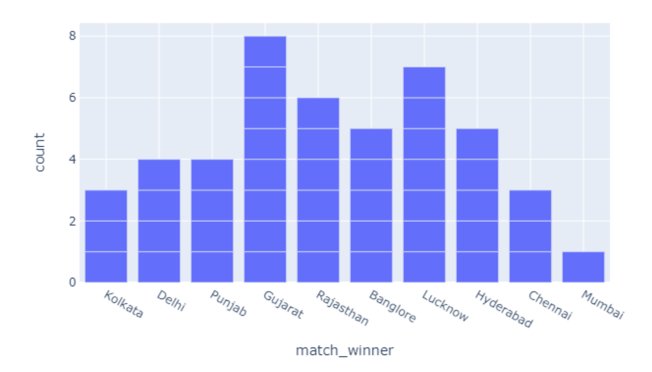
   first_ings_wkts  second_ings_score  second_ings_wkts match_winner    won_by  \
0                5                133                 4      Kolkata  Wickets
1                5                179                 6        Delhi  Wickets
2                2                208                 5       Punjab  Wickets
3                6                161                 5      Gujarat  Wickets
4                6                149                 7    Rajasthan     Runs

   margin player_of_the_match     top_scorer  highscore     best_bowling  \
0       6         Umesh Yadav       MS Dhoni         50     Dwayne Bravo
1       4       Kuldeep Yadav   Ishan Kishan         81    Kuldeep Yadav
2       5         Odean Smith  Faf du Plessis        88   Mohammed Siraj
3       5      Mohammed Shami    Deepak Hooda         55   Mohammed Shami
4      61        Sanju Samson   Aiden Markram         57  Yuzvendra Chahal

  best_bowling_figure
0               3--20
1               3--18
2               2--59
3               3--25
4               3--22
```

تحتوي مجموعة البيانات على جميع المعلومات اللازمة لتلخيص قصة IPL 2022 حتى الآن. لنبدأ بالنظرفي عدد المباريات التي فاز بهاكل فريقفي IPL 2022:

```
figure = px.bar(data, x=data["match_winner”],
            title="Number of Matches Won in IPL 2022”)
figure.show()
```

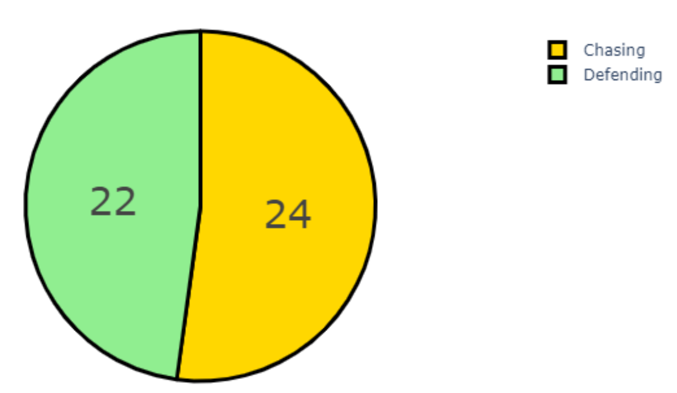


لذا، حاليًا، يقود Gujrat البطولة بفوزه بثماني مباريات. إنه إنجاز كفريق جديد لـ Gujratفي IPL. الآن دعونا نرى كيف تفوز معظم الفرق. سنحلل هنا ما إذا كانت معظم الفرق قد فازت بالدفاع (الضرب أولاً) أو المطاردة (الضربة الثانية):

```
data["won_by"] = data["won_by"].map({"Wickets": "Chasing ,"
"                                      Runs": "Defending”})
won_by = data["won_by"].value_counts()
label = won_by.index
counts = won_by.values
colors = ['gold','lightgreen']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Number of Matches Won By
Defending Or Chasing')
fig.update_traces(hoverinfo='label+percent',
textinfo='value ,'
                  textfont_size=30,
                  marker=dict(colors=colors ,
                              line=dict(color='black',
width=3)))
fig.show()
```

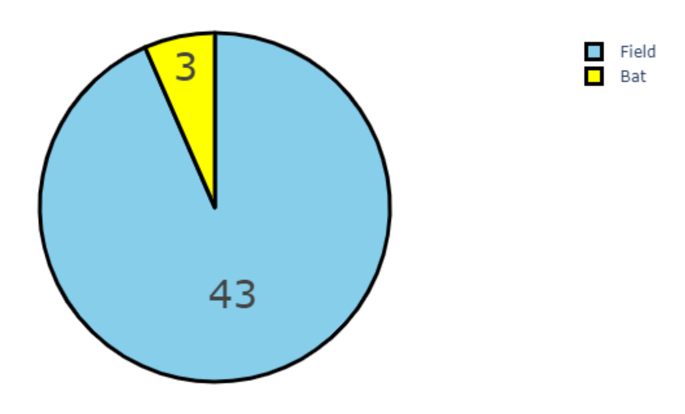




لذلك، حاليًا، تم ربح 24 مباراة أثناء مطاردة الهدف، و 22 مباراة فازت أثناء الدفاع عن الهدف.
الآن دعنا نرى ما تفضله معظم الفرق (الضرب بالملاعب batting or fielding) بعد الفوزفي القرعة:

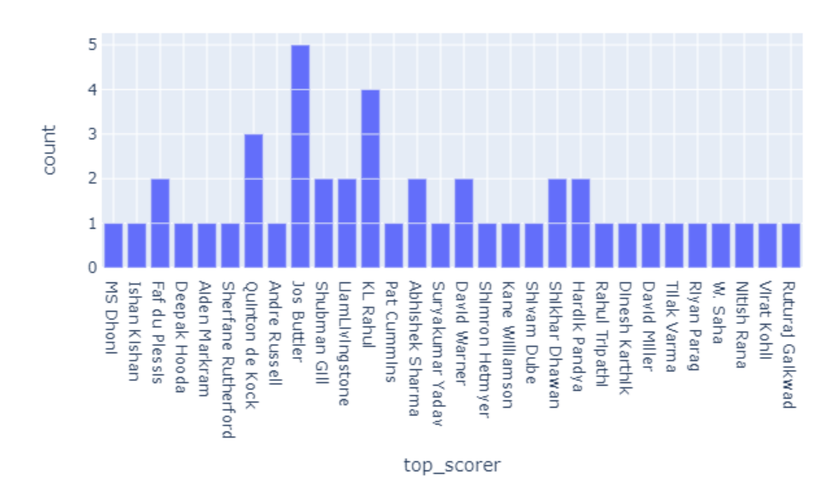
```
 toss = data["toss_decision"].value_counts()
label = toss.index
counts = toss.values
colors = ['skyblue','yellow']

fig = go.Figure(data=[go.Pie(labels=label, values=counts)])
fig.update_layout(title_text='Toss Decision')
fig.update_traces(hoverinfo='label+percent ,'
                  textinfo='value', textfont_size=30,
                  marker=dict(colors=colors ,
                              line=dict(color='black',
width=3)))
fig.show()
```



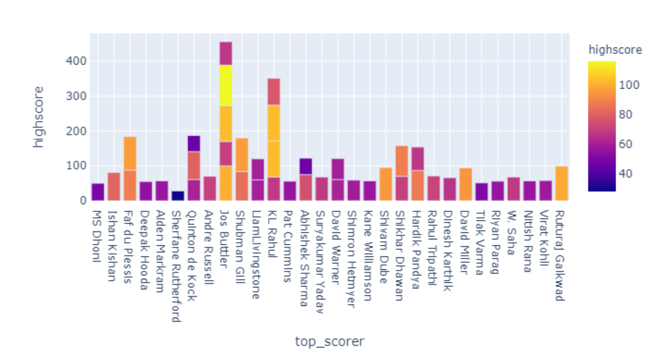
وهكذا، يختار معظم القادة اللعب بعد الفوزفي القرعة. حتى الآن،في 43 مباراة، اختار الكابتن اللعب أولاً، وفي ثلاث مباريات فقط، اختار القادة المضرب أولاً. الآن دعونا نرى أفضل الهدافين في معظم مباريات IPL 2022:

```
figure = px.bar(data, x=data["top_scorer"],
            title="Top Scorers in IPL 2022")
figure.show()
```



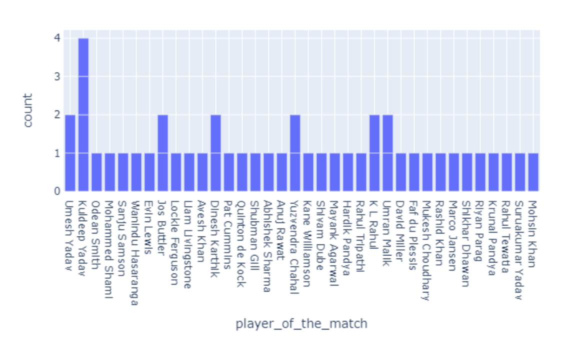
حاليًا، كان Jos Buttler هدافًافي 5 مباريات. إنه يبحثفي لمسة رائعة. دعنا نحللها بعمق من خلال تضمين الأشواط التي سجلها أفضل الهدافين:

```
figure = px.bar(data, x=data["top_scorer"],
                y = data["highscore"] ,
                color = data["highscore"],
            title="Top Scorers in IPL 2022")
figure.show()
```



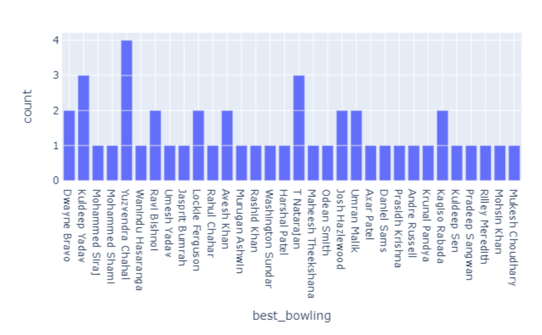
حتى الآن، سجل Jos Buttler ثلاثة قرون، وسجل KL Rahul قرنين من الزمان. الآن دعنا نلقي نظرة على أكثر لاعب من جوائز المباراة حتى الآنفي IPL 2022:

```
figure = px.bar(data, x = data["player_of_the_match”],
                title="Most Player of the Match Awards”)
figure.show()
```



لذا يتصدر Kuldeep Yadav قائمة لاعبي جوائز المباراة بأربع مباريات. إنها بطولة رائعة لـ Kuldeep Yadav هذا العام. الآن دعونا نلقي نظرة على bowlers مع أفضل شخصيات bowlersفي معظم المباريات:

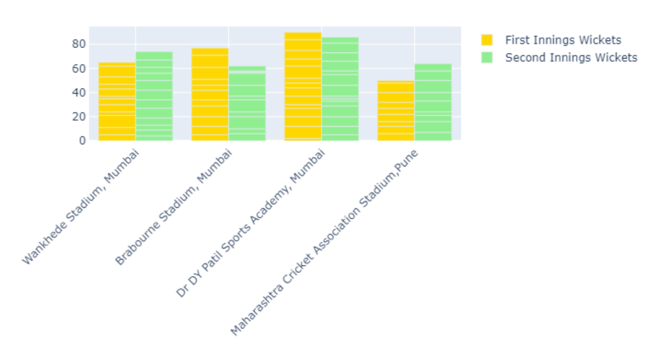
```
figure = px.bar(data, x=data["best_bowling”],
            title="Best Bowlers in IPL 2022”)
figure.show()
```



يمكنك أن ترى Yuzvendra Chahal لديه أفضل شخصيات bowlersفي أربع مباريات. لذا فهذه بطولة رائعة لـ Yuzvendra Chahal هذا العام أيضًا.

دعنا الآن نلقي نظرة على ما إذا كان معظم الويكيت wickets يسقط أثناء تحديد الهدف أو أثناء مطاردة الهدف:

```
figure = go.Figure()
figure.add_trace(go.Bar)
  x=data["venue"],
  y=data["first_ings_wkts"],
  name='First Innings Wickets,'
  marker_color='gold'
))
figure.add_trace(go.Bar)
  x=data["venue"],
  y=data["second_ings_wkts"],
  name='Second Innings Wickets,'
  marker_color='lightgreen'
))
figure.update_layout(barmode='group', xaxis_tickangle=-45)
figure.show()
```



لذلك في استاد Wankhede في مومباي واستاد MCA في Pune ، تسقط معظم الويكيت أثناء مطاردة الهدف. وفي الملعبين الآخرين، تسقط معظم الويكيت أثناء تحديد الهدف. هذه هي الطريقة التي يمكنك بها تحليل وتلخيص قصة IPL 2022 باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكنك بها أداء مهمة تحليل IPL 2022 باستخدام بايثون. يسير IPL 2022 بشكل رائع بالنسبة إلى Gujrat كفريق جديد هذا العام. لقد كان Jos Buttler و KL Rahul رائعين في التعامل مع عصا الكريكيت bat، وكان Yuzvendra Chahal و Kuldeep Yadav رائعين في التعامل مع الرمي bowl. أتمنى أن تكون قد أحببت هذه المقالة حول تحليل IPL 2022 باستخدام بايثون.

16) تحليل تأثيرات Covid-19 باستخدام بايثون Covid-19 Impacts Analysis using Python

أدى اندلاع Covid-19 إلى الكثير من القيود التي أدت إلى العديد من التأثيرات على الاقتصاد العالمي. تأثرت جميع البلدان تقريبًا سلبًا بارتفاع حالات Covid-19. إذا كنت تريد معرفة كيفية تحليل تأثيرات Covid-19 على الاقتصاد، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل تأثيرات Covid-19 باستخدام بايثون.

تحليل آثار Covid-19 (دراسة حالة)

أثرت الموجة الأولى من Covid-19 على الاقتصاد العالمي حيث لم يكن العالم جاهزًا أبدًا لمواجهة الوباء. وقد أدى ذلك إلى ارتفاع عدد الحالات وارتفاع الوفيات وارتفاع معدلات البطالة وارتفاع معدلات الفقر، مما أدى إلى تباطؤ اقتصادي. هنا، أنت مطالب بتحليل انتشار حالات Covid-19 وجميع آثار Covid-19 على الاقتصاد.

يتم تنزيل مجموعة البيانات التي نستخدمها لتحليل تأثيرات covid-19 من Kaggle. يحتوي على بيانات حول:

1. رمز البلد (the country code).
2. اسم كل الدول (name of all the countries).
3. تاريخ التسجيل (date of the record).
4. مؤشر التنمية البشرية لجميع الدول (Human development index of all the countries).
5. حالات Covid-19 اليومية (Daily covid-19 cases).
6. الوفيات اليومية بسبب Covid-19 (Daily deaths due to covid-19).
7. مؤشر صرامة البلدان (stringency index of the countries).
8. سكان الدول (the population of the countries).
9. نصيب الفرد من الناتج المحلي الإجمالي للدول (GDP per capita of the countries).

يمكنك تنزيل مجموعة البيانات هذه من هنا.

تحليل تأثيرات Covid-19 باستخدام بايثون

لنبدأ مهمة تحليل تأثيرات Covid-19 عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go

data = pd.read_csv("transformed_data.csv")
data2 = pd.read_csv("raw_data.csv")
print(data)
```

```
      CODE     COUNTRY        DATE    HDI        TC        TD       STI  \
0      AFG  Afghanistan  2019-12-31  0.498  0.000000  0.000000  0.000000
1      AFG  Afghanistan  2020-01-01  0.498  0.000000  0.000000  0.000000
2      AFG  Afghanistan  2020-01-02  0.498  0.000000  0.000000  0.000000
3      AFG  Afghanistan  2020-01-03  0.498  0.000000  0.000000  0.000000
4      AFG  Afghanistan  2020-01-04  0.498  0.000000  0.000000  0.000000
...    ...          ...         ...    ...       ...       ...       ...
50413  ZWE     Zimbabwe  2020-10-15  0.535  8.994048  5.442418  4.341855
50414  ZWE     Zimbabwe  2020-10-16  0.535  8.996528  5.442418  4.341855
50415  ZWE     Zimbabwe  2020-10-17  0.535  8.999496  5.442418  4.341855
50416  ZWE     Zimbabwe  2020-10-18  0.535  9.000853  5.442418  4.341855
50417  ZWE     Zimbabwe  2020-10-19  0.535  9.005405  5.442418  4.341855

             POP    GDPCAP
0      17.477233  7.497754
1      17.477233  7.497754
2      17.477233  7.497754
3      17.477233  7.497754
4      17.477233  7.497754
...          ...       ...
50413  16.514381  7.549491
50414  16.514381  7.549491
50415  16.514381  7.549491
50416  16.514381  7.549491
50417  16.514381  7.549491

[50418 rows x 9 columns]
```

تحتوي البيانات التي نستخدمها على بيانات عن حالات كوفيد ـ 19 وتأثيرها على الناتج المحلي الإجمالي من 31 ديسمبر 2019 إلى 10 أكتوبر 2020.

تحضير البيانات

تحتوي مجموعة البيانات التي نستخدمها هنا على ملفي بيانات. يحتوي أحد الملفات على بيانات أولية، بينما يحتوي الملف الآخر على ملف تم تحويله. لكن يتعين علينا استخدام مجموعتي البيانات لهذه المهمة، حيث يحتوي كل منهما على معلومات مهمة بنفس القدرفي أعمدة مختلفة. لذلك دعونا نلقي نظرة على مجموعتي البيانات واحدة تلو الأخرى:

```
print(data.head())
```

```
  CODE      COUNTRY        DATE    HDI   TC   TD  STI        POP    GDPCAP
0  AFG  Afghanistan  2019-12-31  0.498  0.0  0.0  0.0  17.477233  7.497754
1  AFG  Afghanistan  2020-01-01  0.498  0.0  0.0  0.0  17.477233  7.497754
2  AFG  Afghanistan  2020-01-02  0.498  0.0  0.0  0.0  17.477233  7.497754
3  AFG  Afghanistan  2020-01-03  0.498  0.0  0.0  0.0  17.477233  7.497754
4  AFG  Afghanistan  2020-01-04  0.498  0.0  0.0  0.0  17.477233  7.497754
```

```
print(data2.head())
```

```
  iso_code     location        date  total_cases  total_deaths  \
0      AFG  Afghanistan  2019-12-31          0.0           0.0
1      AFG  Afghanistan  2020-01-01          0.0           0.0
2      AFG  Afghanistan  2020-01-02          0.0           0.0
3      AFG  Afghanistan  2020-01-03          0.0           0.0
4      AFG  Afghanistan  2020-01-04          0.0           0.0

   stringency_index  population  gdp_per_capita  human_development_index  \
0               0.0    38928341        1803.987                    0.498
1               0.0    38928341        1803.987                    0.498
2               0.0    38928341        1803.987                    0.498
3               0.0    38928341        1803.987                    0.498
4               0.0    38928341        1803.987                    0.498

  Unnamed: 9 Unnamed: 10 Unnamed: 11  Unnamed: 12  Unnamed: 13
0      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
1      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
2      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
3      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
4      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
```

بعد الحصول على انطباعات أولية عن مجموعتي البيانات، وجدت أنه يتعين علينا دمج مجموعتي البيانات من خلال إنشاء مجموعة بيانات جديدة. ولكن قبل إنشاء مجموعة بيانات جديدة، دعنا نلقي نظرة على عدد عينات كل بلد الموجودةفي مجموعة البيانات:

```
data["COUNTRY"].value_counts()
```

```
Thailand                294
China                   294
Norway                  294
Afghanistan             294
United Arab Emirates    294
                       ...
Tajikistan              172
Comoros                 171
Lesotho                 158
Hong Kong                51
Solomon Islands           4
Name: COUNTRY, Length: 210, dtype: int64
```

لذلك ليس لدينا عدد متساوٍ من العينات لكل بلدفي مجموعة البيانات. دعونا نلقي نظرة على قيمة المنوال (mode value):

```
data["COUNTRY"].value_counts().mode()
```

```
0    294
dtype: int64
```

إذن 294 هي قيمة المنوال. سنحتاج إلى استخدامه لقسمة مجموع جميع العينات المتعلقة بمؤشر التنمية البشرية ونصيب الفرد من الناتج المحلي الإجمالي والسكان. فلنقم الآن بإنشاء مجموعة بيانات جديدة من خلال دمج الأعمدة الضرورية من مجموعتي البيانات:

```
#Aggregating the data

code = data["CODE"].unique().tolist()
country = data["COUNTRY"].unique().tolist()
hdi =[]
tc =[]
td =[]
sti =[]
population = data["POP"].unique().tolist()
gdp =[]

for i in country:
  hdi.append((data.loc[data["COUNTRY"] == i, "HDI"]).sum()/294)
  tc.append((data2.loc[data2["location"] == i,
"total_cases"]).sum())
  td.append((data2.loc[data2["location"] == i,
"total_deaths"]).sum())
  sti.append((data.loc[data["COUNTRY"] == i, "STI"]).sum()/294)
  population.append((data2.loc[data2["location"] == i,
"population"]).sum()/294)

aggregated_data = pd.DataFrame(list(zip(code, country, hdi,
tc, td, sti, population)),
                 columns = ["Country Code", "Country", "HDI”,
"                    Total Cases", "Total Deaths”,
"                    Stringency Index", "Population”)
print(aggregated_data.head())
```

```
  Country Code      Country       HDI  Total Cases  Total Deaths  \
0          AFG  Afghanistan  0.498000    5126433.0      165875.0
1          ALB      Albania  0.600765    1071951.0       31056.0
2          DZA      Algeria  0.754000    4893999.0      206429.0
3          AND      Andorra  0.659551     223576.0        9850.0
4          AGO       Angola  0.418952     304005.0       11820.0

   Stringency Index  Population
0          3.049673   17.477233
1          3.005624   14.872537
2          3.195168   17.596309
3          2.677654   11.254996
4          2.965560   17.307957
```

لم أقم بتضمين عمود نصيب الفرد من الناتج المحلي الإجمالي حتى الآن. لم أجد الأرقام الصحيحة للناتج المحلي الإجمالي للفردفي مجموعة البيانات. لذلك سيكون من الأفضل جمع البيانات يدويًا حول نصيب الفرد من الناتج المحلي الإجماليفي البلدان.

نظرًا لوجود العديد من البلدانفي هذه البيانات، فلن يكون من السهل جمع البيانات يدويًا حول نصيب الفرد من الناتج المحلي الإجماليفي جميع البلدان. لذلك دعونا نحدد عينة فرعية من مجموعة البيانات هذه. لإنشاء عينة فرعية من مجموعة البيانات هذه، سأختار أفضل 10 دول بها أكبر عدد من حالات Covid-19. ستكون عينة مثالية لدراسة الآثار الاقتصادية لفيروس كورونا. لذلك دعونا نفرز البيانات وفقًا لإجمالي حالات Covid-19 :

```
#Sorting Data According to Total Cases

data = aggregated_data.sort_values(by=["Total Cases"],
ascending=False)
print(data.head())
```

```
    Country Code        Country      HDI  Total Cases  Total Deaths  \
200          USA  United States  0.92400  746014098.0    26477574.0
27           BRA         Brazil  0.75900  425704517.0    14340567.0
90           IND          India  0.64000  407771615.0     7247327.0
157          RUS         Russia  0.81600  132888951.0     2131571.0
150          PER           Peru  0.59949   74882695.0     3020038.0

     Stringency Index  Population
200          3.350949   19.617637
27           3.136028   19.174732
90           3.610552   21.045353
157          3.380088   18.798668
150          3.430126   17.311165
```

الآن إليك كيف يمكننا تحديد أفضل 10 بلدان بها أكبر عدد من الحالات:

```
#Top 10 Countries with Highest Covid Cases

data = data.head(10)
print(data)
```

```
    Country Code         Country       HDI  Total Cases  Total Deaths  \
200          USA   United States  0.924000  746014098.0    26477574.0
27           BRA          Brazil  0.759000  425704517.0    14340567.0
90           IND           India  0.640000  407771615.0     7247327.0
157          RUS          Russia  0.816000  132888951.0     2131571.0
150          PER            Peru  0.599490   74882695.0     3020038.0
125          MEX          Mexico  0.774000   74347548.0     7295850.0
178          ESP           Spain  0.887969   73717676.0     5510624.0
175          ZAF    South Africa  0.608653   63027659.0     1357682.0
42           COL        Colombia  0.581847   60543682.0     1936134.0
199          GBR  United Kingdom  0.922000   59475032.0     7249573.0

     Stringency Index  Population
200          3.350949   19.617637
27           3.136028   19.174732
90           3.610552   21.045353
157          3.380088   18.798668
150          3.430126   17.311165
125          3.019289   18.674802
178          3.393922   17.660427
175          3.364333   17.898266
42           3.357923   17.745037
199          3.353883   18.033340
```

سأضيف الآن عمودين آخرين (نصيب الفرد من الناتج المحلي الإجمالي قبل Covid-19 ، نصيب الفرد من الناتج المحلي الإجمالي خلال Covid-19) إلى مجموعة البيانات هذه:

```
data["GDP Before Covid"] = [65279.53, 8897.49, 2100.75,
,9946.03 ,7027.61 ,11497.65
[42354.41 ,6424.98 ,6001.40 ,29564.74
data["GDP During Covid"] = [63543.58, 6796.84, 1900.71,
,8346.70 ,6126.87 ,10126.72
[40284.64 ,5332.77 ,5090.72 ,27057.16
print(data)
```

```
    Country Code         Country       HDI  Total Cases  Total Deaths  \
200          USA   United States  0.924000  746014098.0    26477574.0
27           BRA          Brazil  0.759000  425704517.0    14340567.0
90           IND           India  0.640000  407771615.0     7247327.0
157          RUS          Russia  0.816000  132888951.0     2131571.0
150          PER            Peru  0.599490   74882695.0     3020038.0
125          MEX          Mexico  0.774000   74347548.0     7295850.0
178          ESP           Spain  0.887969   73717676.0     5510624.0
175          ZAF    South Africa  0.608653   63027659.0     1357682.0
42           COL        Colombia  0.581847   60543682.0     1936134.0
199          GBR  United Kingdom  0.922000   59475032.0     7249573.0

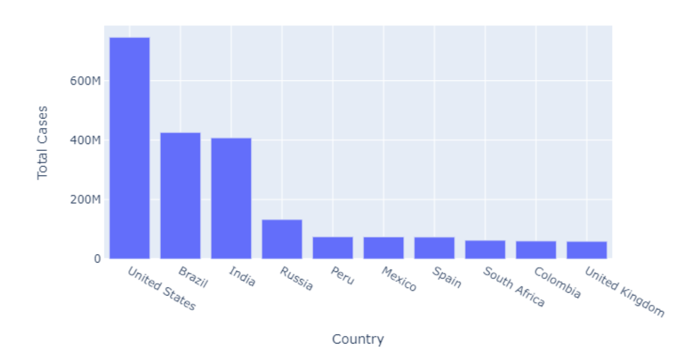
     Stringency Index  Population  GDP Before Covid  GDP During Covid
200          3.350949   19.617637          65279.53          63543.58
27           3.136028   19.174732           8897.49           6796.84
90           3.610552   21.045353           2100.75           1900.71
157          3.380088   18.798668          11497.65          10126.72
150          3.430126   17.311165           7027.61           6126.87
125          3.019289   18.674802           9946.03           8346.70
178          3.393922   17.660427          29564.74          27057.16
175          3.364333   17.898266           6001.40           5090.72
42           3.357923   17.745037           6424.98           5332.77
199          3.353883   18.033340          42354.41          40284.64
```

ملاحظة: يتم جمع البيانات المتعلقة بنصيب الفرد من الناتج المحلي الإجمالي يدويًا.

تحليل انتشار Covid-19

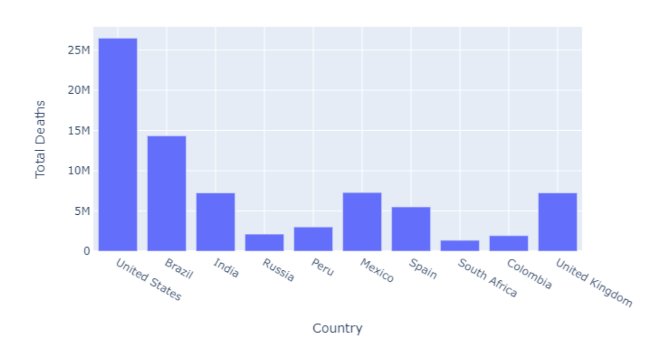
لنبدأ الآن بتحليل انتشار Covid-19 في جميع البلدان التي بها أكبر عدد من حالات الإصابة بفيروس Covid-19. سأقوم أولاً بإلقاء نظرة على جميع البلدان التي بها أكبر عدد من حالات الإصابة بفيروس Covid-19:

```
figure = px.bar(data, y='Total Cases', x='Country',
     title="Countries with Highest Covid Cases")
figure.show()
```



يمكننا أن نرى أن الولايات المتحدة لديها نسبيًا عدد كبير جداً من حالات Covid-19مقارنة بالبرازيل والهندفي المركزين الثاني والثالث. دعنا الآن نلقي نظرة على العدد الإجمالي للوفيات بين البلدان التي بها أكبر عدد من حالات الإصابة بفيروس Covid-19 :

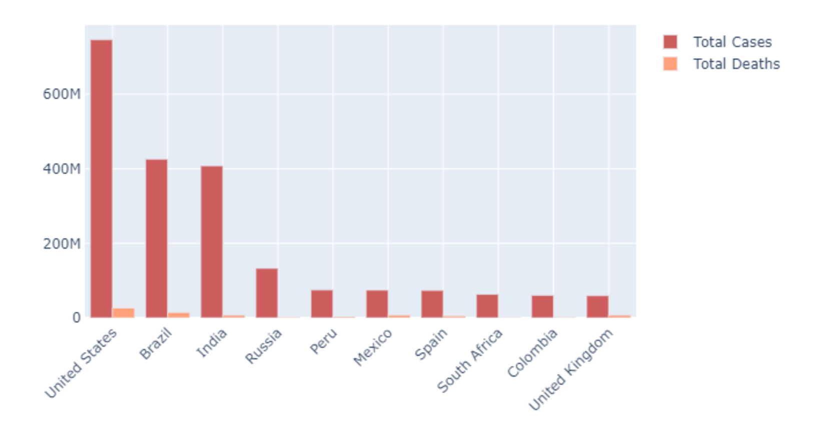
```
figure = px.bar(data, y='Total Deaths', x='Country',
     title="Countries with Highest Deaths")
figure.show()
```



تمامًا مثل العدد الإجمالي لحالات Covid-19، تتصدر الولايات المتحدة عدد الوفيات، مع البرازيل والهندفي المركزين الثاني والثالث. شيء واحد يجب ملاحظته هنا هو أن معدل الوفيات في الهند وروسيا وجنوب إفريقيا منخفض نسبيًا وفقًا للعدد الإجمالي للحالات. دعنا الآن نقارن العدد الإجمالي للحالات وإجمالي الوفياتفي جميع هذه البلدان:

```
fig = go.Figure()
fig.add_trace(go.Bar)
  x=data["Country"],
```

```
  y=data["Total Cases"],
  name='Total Cases',
  marker_color='indianred'
))
fig.add_trace(go.Bar)
  x=data["Country"],
  y=data["Total Deaths"],
  name='Total Deaths',
  marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



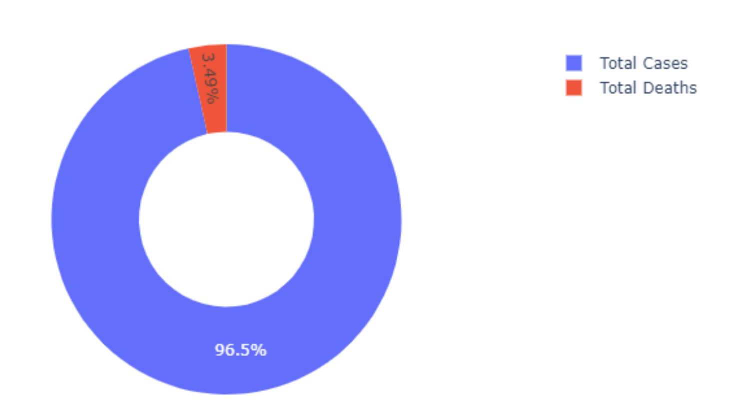
دعونا الآن نلقي نظرة على النسبة المئوية لإجمالي الوفيات وإجمالي الحالات بين جميع البلدان التي بها أكبر عدد من حالات الإصابة بفيروس Covid-19:

```
#Percentage of Total Cases and Deaths
cases = data["Total Cases"].sum()
deceased = data["Total Deaths"].sum()

labels = ["Total Cases", "Total Deaths"]
values = [cases, deceased]

fig = px.pie(data, values=values, names=labels,
     title='Percentage of Total Cases and Deaths', hole=0.5)
fig.show()
```





فيما يلي كيفية حساب معدل الوفيات لحالات Covid-19:

```
death_rate = (data["Total Deaths"].sum() / data["Total
Cases"].sum()) * 100
print("Death Rate = ", death_rate)
```

```
Death Rate =  3.6144212045653767
```

عمود آخر مهم في مجموعة البيانات هذه هو فهرس الصرامة (stringency index). إنه مقياس مركب لمؤشرات الاستجابة، بما في ذلك إغلاق المدارس وإغلاق أماكن العمل وحظر السفر. إنه يوضح مدى صرامة الدول في اتباع هذه الإجراءات للسيطرة على انتشار Covid-19 :

```
fig = px.bar(data, x='Country', y='Total Cases',
     hover_data=['Population', 'Total Deaths'],
     color='Stringency Index', height=400,
     title= "Stringency Index during Covid-19")
fig.show()
```

هنا يمكننا أن نرى أن الهند تؤدي أداءً جيداً في مؤشر الصرامة أثناء تفشي فيروس كورونا.

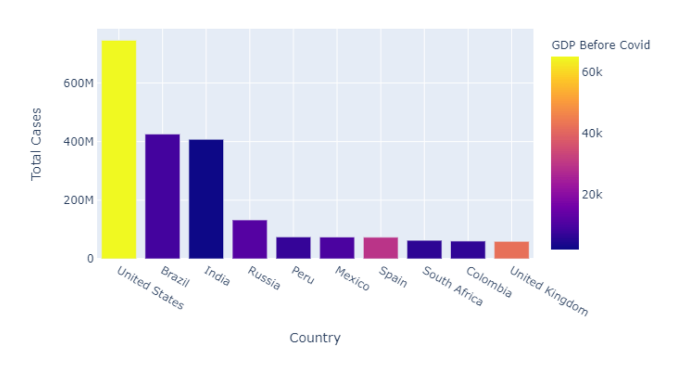
تحليل تأثيرات Covid-19 على الاقتصاد

دعنا الآن ننتقل لتحليل آثار Covid-19 على الاقتصاد. هنا، نصيب الفرد من الناتج المحلي الإجمالي هو العامل الأساسي لتحليل التباطؤ الاقتصادي الناجم عن تفشي Covid-19. دعونا نلقي نظرة على نصيب الفرد من الناتج المحلي الإجمالي قبل تفشي Covid-19 بين البلدان التي بها أكبر عدد من حالات الإصابة بفيروس Covid-19:

```
fig = px.bar(data, x='Country', y='Total Cases',
     hover_data=['Population', 'Total Deaths'],
```

```
     color='GDP Before Covid', height=400,
     title="GDP Per Capita Before Covid-19")
fig.show()
```

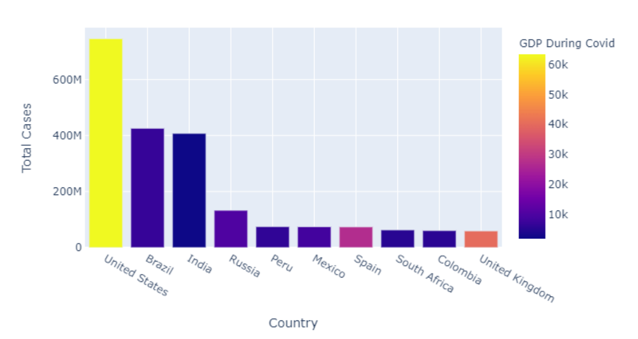




دعونا الآن نلقي نظرة على نصيب الفرد من الناتج المحلي الإجمالي خلال الارتفاع في حالات Covid-19 :

```
fig = px.bar(data, x='Country', y='Total Cases',
     hover_data=['Population', 'Total Deaths'],
     color='GDP During Covid', height=400,
     title="GDP Per Capita During Covid-19")
fig.show()
```

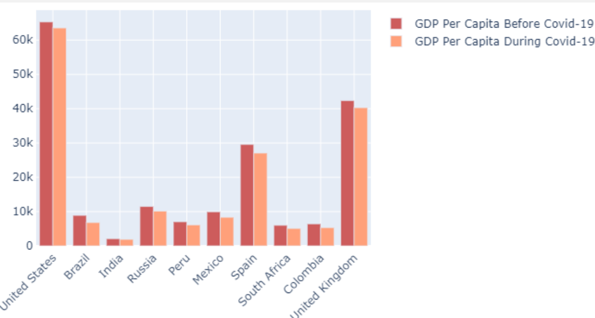
GDP Per Capita During Covid-19



الآن دعونا نقارن نصيب الفرد من الناتج المحلي الإجمالي قبل Covid-19 وأثناء Covid-19 لإلقاء نظرة على تأثير Covid-19 على نصيب الفرد من الناتج المحلي الإجمالي:

```
fig = go.Figure()
fig.add_trace(go.Bar)
  x=data["Country"],
  y=data["GDP Before Covid"],
  name='GDP Per Capita Before Covid-19',
```

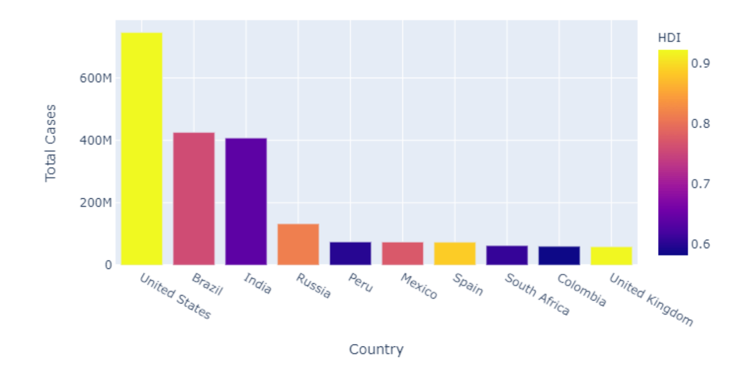
```
  marker_color='indianred'
))
fig.add_trace(go.Bar)
  x=data["Country"],
  y=data["GDP During Covid"],
  name='GDP Per Capita During Covid-19',
  marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



يمكنك أن ترى انخفاضًافي نصيب الفرد من الناتج المحلي الإجماليفي جميع البلدان التي بها أكبر عدد من حالات الإصابة بفيروس كورونا.

عامل اقتصادي مهم آخر هو مؤشر التنمية البشرية (Human Development Index). إنه مؤشر إحصائي مركب لمتوسط العمر المتوقع والتعليم ومؤشرات نصيب الفرد. دعونا نلقي نظرة على عدد البلدان التي كانت تنفق ميزانيتها على التنمية البشرية:

```
fig = px.bar(data, x='Country', y='Total Cases,'
      hover_data=['Population', 'Total Deaths'],
      color='HDI', height=400,
      title="Human Development Index during Covid-19")
fig.show()
```



إذن هذه هي الطريقة التي يمكننا بها تحليل انتشار Covid-19 وتأثيره على الاقتصاد.

الملخص

في هذه المهمة، درسنا انتشار فيروس كورونا بين الدول وتأثيره على الاقتصاد العالمي. لقد رأينا أن تفشي Covid-19 أدى إلى أكبر عدد من حالات الإصابة والوفيات بفيروس Covid-19 في الولايات المتحدة. أحد الأسباب الرئيسية وراء ذلك هو مؤشر الصرامةفي الولايات المتحدة. إنها منخفضة نسبيًا وفقًا لعدد السكان. قمنا أيضًا بتحليل كيفية تأثر نصيب الفرد من الناتج المحلي الإجماليفي كل بلد أثناء تفشي فيروس كورونا. آمل أن تكون قد أحببت هذه المقالة حول تحليل تأثيرات Covid-19 باستخدام بايثون.

16) تحليل مدى الوصول إلى Instagram باستخدام بايثون Instagram Reach Analysis using Python

يعد Instagram أحد أشهر تطبيقات الوسائط الاجتماعية اليوم. يستخدمه الأشخاص الذين يستخدمون Instagram بشكل احترافي للترويج لأعمالهم وبناء محفظة وتدوين وإنشاء أنواع مختلفة من المحتوى. نظرًا لأن Instagram هو تطبيق شائع يستخدمه ملايين الأشخاص من مختلف المجالات، فإن Instagram يستمرفي التغيير لتحسين نفسه لمنشئي المحتوى والمستخدمين. ولكن مع استمرار هذا التغيير، فإنه يؤثر على مدى وصول منشوراتنا التي تؤثر علينا على المدى الطويل. لذلك إذا أراد منشئ المحتوى أن يعمل بشكل جيد على Instagram على المدى الطويل، فعليه أن ينظر إلى بيانات وصوله إلى Instagram. هذا هو المكان الذي يأتي فيه استخدام علم البياناتفي وسائل التواصل الاجتماعي. إذاكنت تريد معرفة كيفية استخدام بيانات Instagram الخاصة بنا لمهمة تحليل الوصول إلى Instagram، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك عبر تحليل الوصول الى Instagramباستخدام بايثون، مما سيساعد منشئي المحتوى على فهم كيفية التكيف مع التغييراتفي Instagram على المدى الطويل.

تحليل مدى الوصول إلى Instagram

لقد كنت أبحث عن الوصول إلى Instagram لفترة طويلة الآن.في كل مرة أنشر فيها على حساب Instagram الخاص بي، أقوم بجمع بيانات حول مدى وصول المنشور بعد أسبوع. يساعد ذلك في فهم كيفية عمل خوارزمية Instagram. إذا كنت ترغبفي تحليل مدى وصول حساب Instagram الخاص بك، فيجب عليك جمع بياناتك يدويًا حيث توجد بعض واجهات برمجة التطبيقات API، لكنها لا تعمل بشكل جيد. لذلك من الأفضل جمع بيانات Instagram يدويًا.

إذاكنت طالبًافي علم البيانات وترغبفي تعلم تحليل الوصول إلى Instagram باستخدام بايثون، فيمكنك استخدام البيانات التي جمعتها من حسابي على Instagram. يمكنك تنزيل مجموعة البيانات التي استخدمتها لمهمة تحليل مدى الوصول إلى Instagram من هنا. الآنفي القسم أدناه، سوف آخذك خلال مهمة تحليل الوصول إلى Instagram والتنبؤ باستخدام التعلم الآلي باستخدام بايثون.

تحليل مدى الوصول إلى Instagram باستخدام بايثون

لنبدأ الآن مهمة تحليل مدى وصول حساب Instagram الخاص بي عن طريق استيراد مكتبات Python ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveRegressor

data = pd.read_csv("Instagram.csv", encoding = 'latin1')
print(data.head())
```

```
   Impressions  From Home  From Hashtags  From Explore  From Other  Saves  \
0       3920.0     2586.0         1028.0         619.0        56.0   98.0
1       5394.0     2727.0         1838.0        1174.0        78.0  194.0
2       4021.0     2085.0         1188.0           0.0       533.0   41.0
3       4528.0     2700.0          621.0         932.0        73.0  172.0
4       2518.0     1704.0          255.0         279.0        37.0   96.0

   Comments  Shares  Likes  Profile Visits  Follows  \
0       9.0     5.0  162.0            35.0      2.0
1       7.0    14.0  224.0            48.0     10.0
2      11.0     1.0  131.0            62.0     12.0
3      10.0     7.0  213.0            23.0      8.0
4       5.0     4.0  123.0             8.0      0.0

                                             Caption  \
0  Here are some of the most important data visua...
1  Here are some of the best data science project...
2  Learn how to train a machine learning model an...
3  Here�s how you can write a Python program to d...
4  Plotting annotations while visualizing your da...

                                            Hashtags
0  #finance #money #business #investing #investme...
1  #healthcare #health #covid #data #datascience ...
2  #data #datascience #dataanalysis #dataanalytic...
3  #python #pythonprogramming #pythonprojects #py...
4  #datavisualization #datascience #data #dataana...
```

قبل البدءفي كل شيء، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
Impressions       1
From Home         1
From Hashtags     1
From Explore      1
From Other        1
Saves             1
Comments          1
Shares            1
Likes             1
Profile Visits    1
Follows           1
Caption           1
Hashtags          1
dtype: int64
```

لذلك فهي تحتوي على قيمة فارغةفي كل عمود. دعنا نسقط كل هذه القيم الفارغة ونتحرك إلى أبعد من ذلك:

```
data = data.dropna()
```

دعنا نلقي نظرة على رؤى الأعمدة لفهم نوع البيانات لجميع الأعمدة:

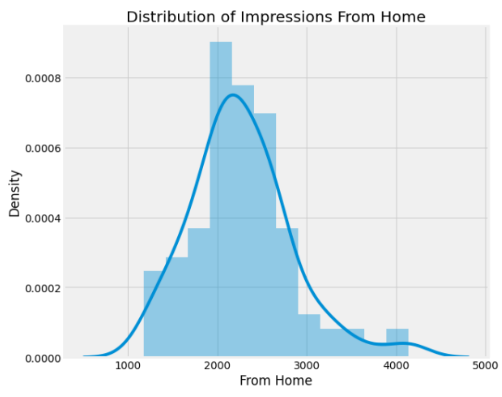
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 99 entries, 0 to 98
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Impressions     99 non-null     float64
 1   From Home       99 non-null     float64
 2   From Hashtags   99 non-null     float64
 3   From Explore    99 non-null     float64
 4   From Other      99 non-null     float64
 5   Saves           99 non-null     float64
 6   Comments        99 non-null     float64
 7   Shares          99 non-null     float64
 8   Likes           99 non-null     float64
 9   Profile Visits  99 non-null     float64
 10  Follows         99 non-null     float64
 11  Caption         99 non-null     object
 12  Hashtags        99 non-null     object
dtypes: float64(11), object(2)
memory usage: 10.8+ KB
```

تحليل مدى وصول Instagram

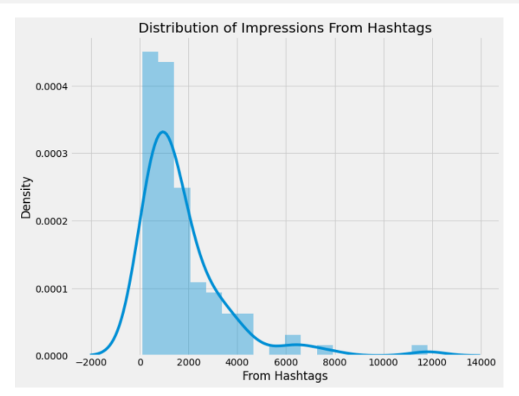
لنبدأ الآن بتحليل مدى وصول مشاركاتي على Instagram. سألقي نظرة أولاً على توزيع الانطباعات (distribution of impressions) التي تلقيتها من قسم الصفحة الرئيسية (home).

```
plt.figure(figsize=(10, 8))
plt.style.use('fivethirtyeight')
plt.title("Distribution of Impressions From Home")
sns.distplot(data['From Home'])
plt.show()
```



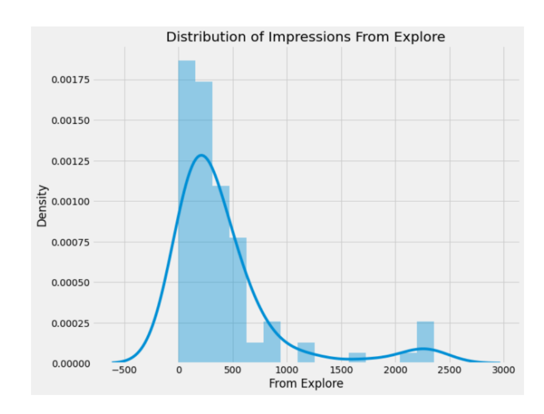
تُظهر الانطباعات التي أحصل عليها من قسم الصفحة الرئيسية على Instagram مدى وصول مشاركاتي إلى متابعيني. بالنظر إلى الانطباعات من قسم الصفحة الرئيسية، يمكنني القول إنه من الصعب الوصول إلى كل متابعيني يوميًا. دعنا الآن نلقي نظرة على توزيع مرات الظهور التي تلقيتها من الوسوم (hashtags):

```
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Hashtags")
sns.distplot(data['From Hashtags'])
plt.show()
```



الوسوم هي أدوات نستخدمها لتصنيف منشوراتنا على Instagram حتى نتمكن من الوصول إلى المزيد من الأشخاص بناءً على نوع المحتوى الذي نقوم بإنشائه. يُظهر النظر إلى مرات ظهور علامة التصنيف أنه لا يمكن الوصول إلى جميع المنشورات باستخدام الوسوم، ولكن يمكن الوصول إلى العديد من المستخدمين الجدد من خلال الوسوم. دعنا الآن نلقي نظرة على توزيع مرات الظهور التي تلقيتها من قسم الاستكشاف (explore section)في Instagram :

```
plt.figure(figsize=(10, 8))
plt.title("Distribution of Impressions From Explore")
sns.distplot(data['From Explore'])
plt.show()
```



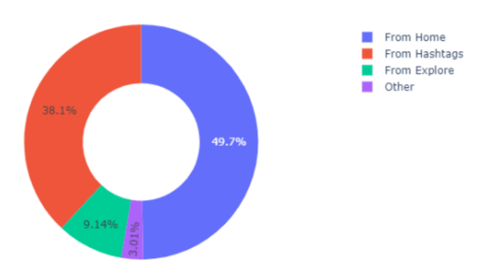
قسم الاستكشاففي Instagram هو نظام التوصيةفي Instagram .توصي المنشورات للمستخدمين بناءً على تفضيلاتهم واهتماماتهم. من خلال النظر إلى الانطباعات التي تلقيتها من قسم الاستكشاف، يمكنني القول أن Instagram لا يوصى كثيرًا بمنشوراتنا للمستخدمين. تلقت بعض المشاركات وصولاً جيدًا من قسم الاستكشاف، لكنها لا تزال منخفضة جدًا مقارنةً بمدى الوصول الذي أتلقاه من الوسوم.

دعنا الآن نلقي نظرة على النسبة المئوية للانطباعات التي أحصل عليها من مصادر مختلفة على Instagram:

```
home = data["From Home"].sum()
hashtags = data["From Hashtags"].sum()
explore = data["From Explore"].sum()
other = data["From Other"].sum()

labels = ['From Home','From Hashtags','From Explore','Other']
values = [home, hashtags, explore, other]

fig = px.pie(data, values=values, names=labels,
      title='Impressions on Instagram Posts From Various
Sources', hole=0.5)
fig.show()
```



لذا تُظهر مخطط الدونات (donut plot) أعلاه أن ما يقرب من 50في المائة من مدى الوصول من متابعيني، و38.1في المائة من الوسوم، و9.14في المائة من قسم الاستكشاف، و3.01في المائة من مصادر أخرى.

تحليل المحتوى

الآن دعنا نحلل محتوى مشاركاتي على Instagram. تحتوي مجموعة البيانات على عمودين، وهما التسمية التوضيحية (caption) والوسوم (hashtags)، والتي ستساعدنافي فهم نوع المحتوى الذي أنشره على Instagram.

دعنا ننشئ سحابة الكلمات (wordcloud) لعمود التسمية التوضيحية لإلقاء نظرة على الكلمات الأكثر استخدامًافي التسمية التوضيحية لمشاركاتي على Instagram:

```
text = " ".join(i for i in data.Caption)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.style.use('classic')
plt.figure( figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



الآن، دعنا ننشئ سحابة الكلمات لعمود الوسوم لإلقاء نظرة على الوسوم الأكثر استخدامًافي مشاركاتي على Instagram:

```
text = " ".join(i for i in data.Hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(12,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

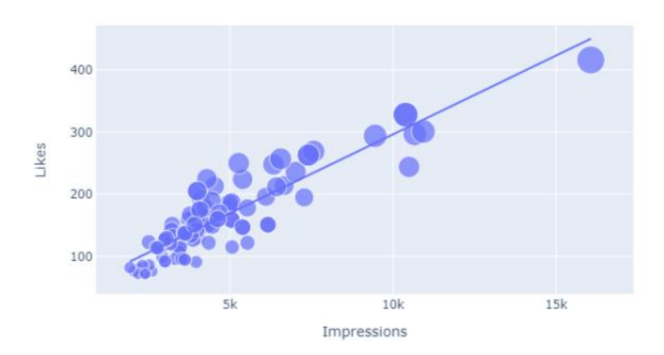
تحليل العلاقات

دعنا الآن نحلل العلاقات للعثور على أهم عوامل وصولنا إلى Instagram سيساعدنا أيضًافي فهم كيفية عمل خوارزميةInstagram .

دعونا نلقي نظرة على العلاقة بين عدد الإعجابات وعدد مرات الانطباعات على مشاركاتي على Instagram :

```
figure = px.scatter(data_frame = data, x="Impressions",
        y="Likes", size="Likes", trendline="ols,"
        title = "Relationship Between Likes and Impressions")
figure.show()
```

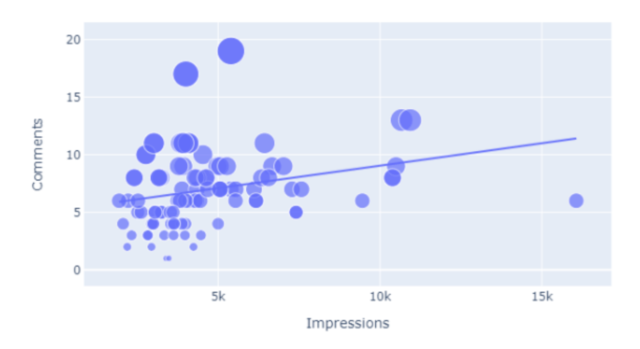
Relationship Between Likes and Impressions



هناك علاقة خطية بين عدد الإعجابات ومدى الوصول الذي حصلت عليه على Instagram
دعنا الآن نرى العلاقة بين عدد التعليقات (comments) وعدد مرات الانطباعات (Impressions) على مشاركاتي علىInstagram :

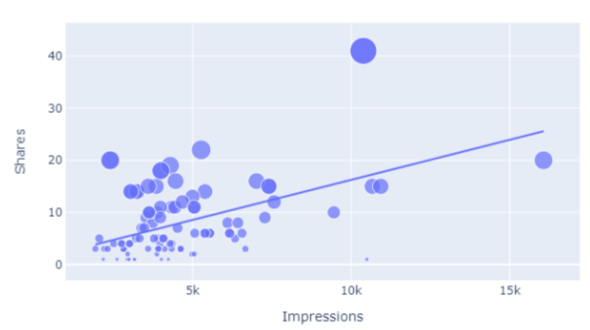
```
figure = px.scatter(data_frame = data, x="Impressions",
        y="Comments", size="Comments", trendline="ols",
        title = "Relationship Between Comments and Total
Impressions")
figure.show()
```

Relationship Between Comments and Total Impressions



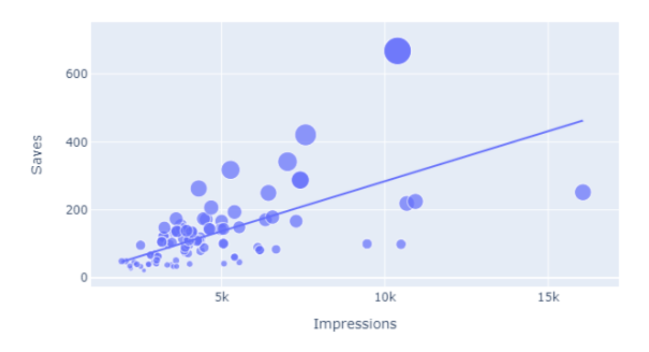
يبدو أن عدد التعليقات التي نتلقاها على إحدى المشاركات (shares) لا يؤثرفي مدى وصولها. دعنا الآن نلقي نظرة على العلاقة بين عدد المشاركات وعدد مرات الظهور:

```
figure = px.scatter(data_frame = data, x="Impressions",
        y="Shares", size="Shares", trendline="ols",
        title = "Relationship Between Shares and Total
Impressions")
figure.show()
```



سيؤدي عدد أكبر من المشاركات إلى وصول أعلى، لكن المشاركات لا تؤثر على مدى وصول المنشور بقدر ما تؤثر إبداءات الإعجاب. دعنا الآن نلقي نظرة على العلاقة بين عدد مرات الحفظ (saves) وعدد مرات الانطباعات (impressions):

```
figure = px.scatter(data_frame = data, x="Impressions",
        y="Saves", size="Saves", trendline="ols",
        title = "Relationship Between Post Saves and Total
Impressions",
figure.show()
```



هناك علاقة خطية بين عدد المرات التي يتم فيها حفظ المنشور الخاص بي ومدى وصول منشور Instagram الخاص بي. دعنا الآن نلقي نظرة على ارتباط جميع الأعمدة بعمود مرات الانطباعات (Impressions column):

```
correlation = data.corr()
print(correlation["Impressions"].sort_values(ascending=False))
```

```
Impressions       1.000000
Likes             0.896277
From Hashtags     0.892682
Follows           0.804064
Profile Visits    0.774393
Saves             0.625600
From Home         0.603378
From Explore      0.498389
Shares            0.476617
From Other        0.429227
Comments          0.247201
Name: Impressions, dtype: float64
```

لذلك يمكننا القول أن المزيد من الإعجابات والحفظ سيساعدكفي الوصول إلى المزيد على Instagram سيساعدك العدد الأكبر من المشاركات أيضًافي الوصول إلى المزيد، لكن العدد المنخفض من المشاركات لن يؤثر على وصولك أيضًا.

تحليل معدل التحول

في Instagram ، معدل المحادثة (conversation rate) يعني عدد المتابعين الذين تحصل عليهم من عدد زيارات الملف الشخصي (total profile visits) من المنشور. الصيغة التي يمكنك استخدامها لحساب معدل التحويل هي (المتابعات / زيارات الملف الشخصي) * 100.

الآن دعنا نلقي نظرة على معدل المحادثةفي حسابي علىInstagram :

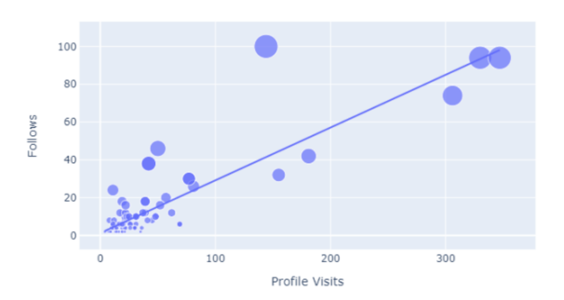
```
conversion_rate = (data["Follows"].sum() / data["Profile
Visits"].sum()) * 100
print(conversion_rate)
```

```
31.17770767613039
```

لذا فإن معدل المحادثةفي حساب Instagram الخاص بي هو 31٪ وهو ما يبدو وكأنه معدل محادثة جيد جداً. دعونا نلقي نظرة على العلاقة بين إجمالي زيارات الملف الشخصي وعدد المتابعين المكتسبين من جميع زيارات الملف الشخصي:

```
figure = px.scatter(data_frame = data, x="Profile Visits”,
        y="Follows", size="Follows", trendline="ols”,
        title = "Relationship Between Profile Visits and
Followers Gained”)
figure.show()
```





العلاقة بين زيارات الملف الشخصي والمتابعين المكتسبة هي أيضًا علاقة خطية.

نموذج توقع الوصول إلى Instagram

الآنفي هذا القسم، سأقوم بتدريب نموذج التعلم الآلي للتنبؤ بمدى وصول منشور على Instagram دعنا نقسم البيانات إلى مجموعات تدريب واختبار قبل تدريب النموذج:

```
x = np.array(data[['Likes', 'Saves', 'Comments', 'Shares',
'        Profile Visits', 'Follows']])
y = np.array(data["Impressions"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                             test_size=0.2,
                             random_state=42)
```

إليك الآن كيف يمكننا تدريب نموذج التعلم الآلي على توقع مدى وصول منشور على Instagram باستخدام بايثون:

```
model = PassiveAggressiveRegressor()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

```
array([10319.5922441])
```

دعنا الآن نتوقع مدى وصول منشور على Instagram من خلال تقديم مدخلات لنموذج التعلم الآلي:

```
#Features = [['Likes','Saves', 'Comments', 'Shares', 'Profile
Visits', 'Follows']]
features = np.array([[54.0,165.0,9.0,4.0,233.0,282.0]])
model.predict(features)
```

```
array([10319.5922441])
```

الملخص

هذه هي الطريقة التي يمكنك بها تحليل وتوقع مدى وصول منشورات Instagram باستخدام التعلم الآلي باستخدام بايثون .إذا أراد منشئ المحتوى أن يؤدي أداءً جيداً على Instagram على المدى الطويل، فعليه إلقاء نظرة على بيانات وصوله إلى Instagram .هذا هو المكان الذي يأتي فيه استخدام علم البياناتفي وسائل التواصل الاجتماعي. آمل أن تكون قد أحببت هذه المقالة حول مهمة تحليل الوصول إلى Instagram باستخدام بايثون.

17) تحليل مشاعر مراجعات Tinder باستخدام بايثون

Tinder Reviews Sentiment Analysis using Python

Tinder هو أحد أكثر تطبيقات (dating) المواعدة شيوعًا. يربط الأشخاص الذين لديهم اهتمامات مماثلة. للتحقق مما إذا كان Tinder يساعد الأشخاص في العثور على شركاء، يمكننا تحليل مشاعر الناس حول Tinder. هناك الكثير من المراجعات على متجر Google Play حول Tinder. يمكننا استخدام هذه البيانات لتحليل مشاعر مستخدمي Tinder. لذلك إذا كنت تريد معرفة كيفية تحليل مراجعات Tinder، فهذه المقالة مناسبة لك. في هذه المقالة، سوف آخذك خلال مهمة مراجعة Tinder لتحليل المشاعر باستخدام بايثون.

تحليل مشاعر مراجعات Tinder باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل آراء Tinder من Kaggle. تم جمعها من مراجعات Tinder على متجر Google Play. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة لبدء هذه المهمة:

```
import pandas as pd
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
import re
from nltk.corpus import stopwords
import string
data = pd.read_csv("tinder_google_play_reviews.csv")
print(data.head())
```

```
                                            reviewId      userName  \
0  gp:AOqpTOF5m-nY12XsKXO0IG-ZQtyvmjwKEp43ILLrhBS...    Kreg Smith
1  gp:AOqpTOFMaTJ6Mj-6hrp6ZI9gU5fzeVZQA9LugbFe1xR...          R.W.
2  gp:AOqpTOGtOLC4xZzU1NT8t1ykvQHfOuhW7oJ0MScukLj...  Benjo cantor
3  gp:AOqpTOGcid22sko0XyvhV1kSpbdKUzx5Q1SIi5L1Ovc...   Chris Plata
4  gp:AOqpTOGzA20eNWEOUM8edTHGQfd6OU7Qy48JpUcBT-x...    Dave Midas

                                           userImage  \
0  https://play-lh.googleusercontent.com/a/AATXAJ...
1  https://play-lh.googleusercontent.com/a-/AOh14...
2  https://play-lh.googleusercontent.com/a/AATXAJ...
3  https://play-lh.googleusercontent.com/a/AATXAJ...
4  https://play-lh.googleusercontent.com/a-/AOh14...

                                             content  score  thumbsUpCount  \
0  Got banned for life don't know why they won't ...      1              0
1  I don't know why I was banned .. But I m not a...      1              0
2                  All gays even if your straight🧑‍🦲🧑‍🦲🧑‍🦲      1              0
3  You have to pay so much to even be seen on thi...      1              0
4  I do not understand how so many people use thi...      2              0
```

```
  reviewCreatedVersion                  at replyContent repliedAt
0               13.6.1  2022-05-21 04:10:44          NaN       NaN
1                  NaN  2022-05-21 04:08:24          NaN       NaN
2               13.6.1  2022-05-21 04:00:10          NaN       NaN
3                  NaN  2022-05-21 03:47:58          NaN       NaN
4               13.6.1  2022-05-21 03:47:51          NaN       NaN
```

في الانطباعات الأولى لمجموعة البيانات هذه، يمكنني رؤية بعض القيم الخاليةفي بعض الأعمدة. لتحليل مراجعات Tinder، نحتاج فقط إلى عمود المحتوى (content column). لذلك دعونا ننشئ مجموعة بيانات جديدة مع عمود المحتوى ونتحرك إلى أبعد من ذلك:

```
data = data[["content"]]
```

دعنا الآن نرى ما إذا كانت لدينا قيم خاليةفي عمود المحتوى:

```
data.isnull().sum()
```

يحتوي عمود المحتوى أيضًا على قيم خالية، فلنقم بإزالة القيم الخالية والمضي قدمًا:

```
data = data.dropna()
```

دعنا الآن نجهز هذه البيانات لمهمة تحليل المشاعر (sentiment analysis). هنا يتعين علينا تنظيف النص في عمود المحتوى:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["content"] = data["content"].apply(clean)
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في مراجعات Tinder:

```
text = " ".join(i for i in data.content)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
```

```
plt.show()
```

سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها ايجابية " Positive" وسلبية "Negative" ومحايدة "Neutral" من خلال حساب درجات المشاعر (sentiment scores) للمراجعات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["content"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["content"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["content"]]
data = data[["content", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                             content  Positive  Negative  \
0  got ban life dont know wont tell help way read...     0.231     0.225
1  dont know ban  allow use tinder henceforth did...     0.371     0.108
2                               gay even straight 👱👱👱     0.000     0.000
3  pay much even seen app girl ever talk back bet...     0.229     0.128
4  understand mani peopl use app im mean ugli ive...     0.000     0.225

   Neutral
0    0.544
1    0.521
2    1.000
3    0.642
4    0.775
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في التعليقات الإيجابية على Tinder:

```
positive =' '.join([i for i in
data['content'][data['Positive'] > data["Negative"]]])
```

```
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الأشخاص في المراجعات السلبية لـ Tinder:

```
negative =' '.join([i for i in
data['content'][data['Negative'] > data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا نلقي نظرة على النتيجة الإجمالية لمشاعر مستخدمي Tinder :

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك يكتب معظم المستخدمين تعليقات محايدة. دعونا نلقي نظرة على إجمالي جميع درجات المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  158277.42200002735
Negative:  59438.14199999961
Neutral:  314250.34899997106
```

كما ترى، الإيجابي أكثر بكثير من السلبي، يمكننا القول إن معظم المستخدمين سعداء بـ Tinder.

الملخص

هذه هي الطريقة التي يمكنك بها تنفيذ مهمة Tinder تحليل مشاعر مراجعات Tinder باستخدام بايثون. هو أحد أكثر تطبيقات المواعدة شيوعًا. يربط الأشخاص الذين لديهم اهتمامات مماثلة. آمل أن تكون قد أحببت هذا المقال حول تحليل المراجعات في Tinder.

18) تحليل المشاعر لمراجعات TikTok باستخدام بايثون TikTok Reviews Sentiment Analysis using Python

يعد TikTok أحد أشهر تطبيقات الوسائط الاجتماعية اليوم. تشتهر بمقاطع الفيديو القصيرة الخاصة بها. غالبًا ما يستخدم الأشخاص هذا التطبيق لمشاهدة مقاطع فيديو مسلية ومضحكة. على الرغم من محتواه الترفيهي، إلا أن هذا التطبيق لا يحبه الجميع. إذا تصفحت مراجعاته، فستجد مزيجًا من الكراهية والدعم لـ TikTokفي جميع أنحاء العالم. لذلك، دعنا نحلل مراجعات TikTok لمعرفة ما يشعر به الناس حول محتوى هذا التطبيق.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر لمراجعات TikTok باستخدام بايثون.

تحليل المشاعر لمراجعات TikTok باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها هنا لمهمة مراجعات TikTok لتحليل المشاعر من Kaggle. تم جمعهافي الأصل من تقييمات TikTok على متجر Google Play. يمكنك تنزيل مجموعة البيانات هذه من هنا. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة لبدء هذه المهمة:

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.corpus import stopwords
import string
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")

data = pd.read_csv("tiktok.csv")
print(data.head())
```

```
                                          reviewId        userName  \
0  gp:AOqpTOHRz-11c0apHLSKHHp52FxUXsQS9Z88wP3sWc5...    MR LOL GAMER
1  gp:AOqpTOF6mFDEkIypmyT3shDLjPHg8zB3kdns2W36ahp...     Dino Kljako
2  gp:AOqpTOGtqU4sb8vuVo3-eB7kIXWoBn-0YCUZ1SnPRKS...  Olivia Harding
3  gp:AOqpTOFHDm-Qa5R6jCpOGTFT2qr1_PKbCTbBNPahCEn...         Keli We
4  gp:AOqpTOFB6Ndao8IHRpOJRmbSknwMGxHcwYzux93YyXI...    Mavis Kotoka

                                         userImage  \
0  https://play-lh.googleusercontent.com/a/AATXAJ...
1  https://play-lh.googleusercontent.com/a-/AOh14...
2  https://play-lh.googleusercontent.com/a/AATXAJ...
3  https://play-lh.googleusercontent.com/a-/AOh14...
4  https://play-lh.googleusercontent.com/a/AATXAJ...
```

```
                                              content  score  thumbsUpCount  \
0                                                Good      5              0
1  Awesome app! Too many people on it where it's ...      5              0
2                                             Not bad      5              0
3                                          It is good      2              0
4                                 Very interesting app      5              0

  reviewCreatedVersion                   at replyContent repliedAt
0               23.8.4  2022-04-05 23:18:30          NaN       NaN
1                  NaN  2022-04-05 23:18:21          NaN       NaN
2               23.9.5  2022-04-05 23:17:34          NaN       NaN
3               22.2.5  2022-04-05 23:17:04          NaN       NaN
4               22.1.5  2022-04-05 23:17:04          NaN       NaN
```

في الانطباعات (impressions) الأولى لمجموعة البيانات هذه، يمكنني رؤية قيم خاليةفي بعض الأعمدة. لتحليل تقييمات TikTok ، نحتاج فقط إلى عمودين ، المحتوى (content) والنتيجة (score)؛ لذلك دعونا ننشئ مجموعة بيانات جديدة مع هذين العمودين فقط ونتقدم خطوة أخرى من خلال تحليل آراء TikTok:

```
data = data[["content", "score"]]
print(data.head())
```

```
                                              content  score
0                                                Good      5
1  Awesome app! Too many people on it where it's ...      5
2                                             Not bad      5
3                                          It is good      2
4                                 Very interesting app      5
```

دعنا الآن نرى ما إذا كان أي من هذين العمودين يحتوي على أي قيم فارغة:

```
print(data.isnull().sum())
```

```
content    4
score      0
dtype: int64
```

لذلك هناك أربع قيم فارغةفي عمود المحتوى. دعنا نسقط القيم الفارغة ونتحرك أبعد من ذلك:

```
data = data.dropna()
```

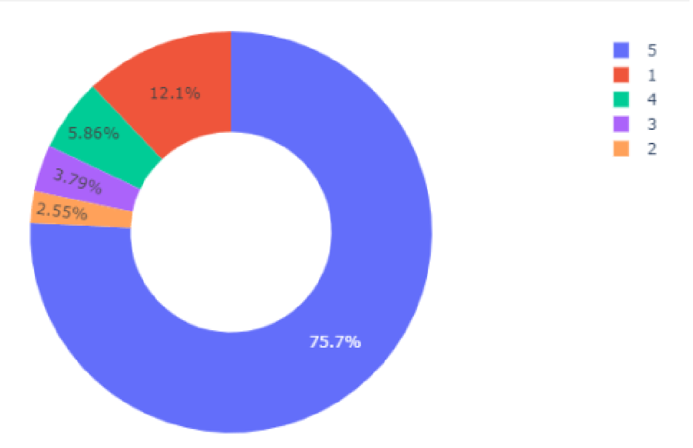
دعنا الآن نجهز هذه البيانات لمهمة تحليل المشاعر. هنا يتعين علينا تنظيف النصفي عمود المحتوى:

```
stopword=set(stopwords.words('english'))
def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
```

```
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["content"] = data["content"].apply(clean)
```

دعنا الآن نلقي نظرة على النسب المئوية للتقييمات الممنوحة لـ TikTok على متجر Google Play :

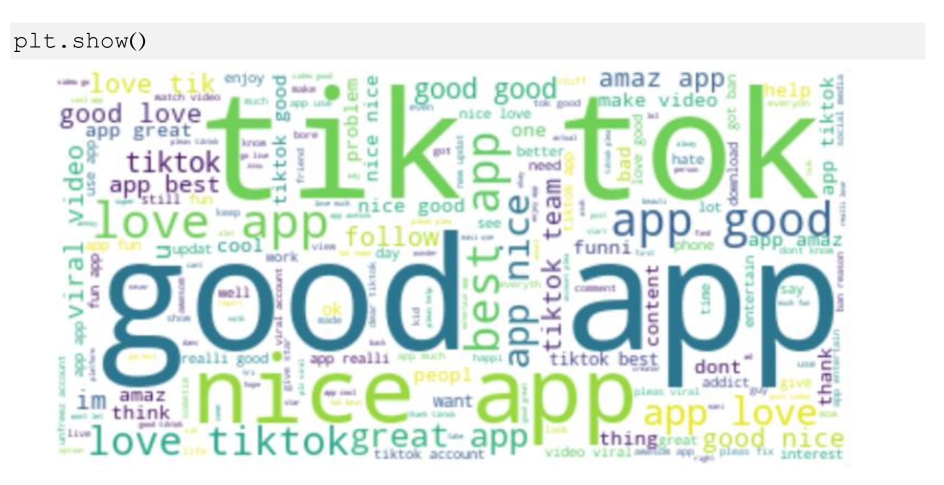
```
ratings = data["score"].value_counts()
numbers = ratings.index
quantity = ratings.values
import plotly.express as px
figure = px.pie(data,
     values=quantity,
     names=numbers,hole = 0.5)
figure.show()
```



يمكنك أن ترى أن ٪75.7 من المستخدمين قد أعطوا خمسة تقييمات لـ TikTok ، و ٪12.1 من المستخدمين صنفوها بـ 1. الآن دعنا نلقي نظرة على نوع الكلمات التي يستخدمها المستخدمونفي مراجعات TikTok :

```
text = " ".join(i for i in data.content)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
```

```
plt.show()
```



سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للتغريدات:

```
                                             content  Positive  Negative  \
0                                               good     1.000       0.0
1  awesom app mani peopl easier fb girl awesom gu...     0.381       0.0
2                                                bad     0.000       1.0
3                                               good     1.000       0.0
4                                       interest app     0.750       0.0

   Neutral
0    0.000
1    0.619
2    0.000
3    0.000
4    0.250
```

دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الناس في التعليقات الإيجابية على TikTok:



دعنا الآن نلقي نظرة على نوع الكلمات التي يستخدمها الناس غالبًافي التعليقات السلبية على :TikTok

```
negative =' '.join([i for i in
data['content'][data['Negative'] > data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات TikTok باستخدام لغة برمجة بايثون.

الملخص

إذن هذه هي كيفية تحليل مشاعر مراجعات TikTok . تشتهر TikTok بمقاطع الفيديو القصيرة الخاصة بها. غالبًا ما يستخدم الأشخاص هذا التطبيق لمشاهدة مقاطع فيديو مسلية ومضحكة. على الرغم من محتواه الترفيهي، إلا أن هذا التطبيق لا يحبه الجميع. آمل أن تكون قد أحببت هذه المقالة حول تحليل مشاعر مراجعات TikTok باستخدام بايثون.

19) تحليل مشاعر حرب أوكرانيا وروسيا على تويتر باستخدام بايثون Ukraine Russia War Twitter Sentiment Analysis using Python

اليوم هو اليوم التاسع عشر للحرب بين روسيا وأوكرانيا. تدعم العديد من الدول أوكرانيا من خلال فرض عقوبات اقتصادية على روسيا. هناك الكثير من التغريدات حول حرب أوكرانيا وروسيا حيث يميل الناس إلى تحديث الحقائق على الأرض، وما يشعرون به حيال ذلك، ومن يدعمون. لذلك إذا كنت ترغب في تحليل مشاعر الناس بشأن الحرب في أوكرانيا وروسيا، فهذا المقال مناسب لك. في هذه المقالة، سأطلعك على مهمة تحليل مشاعر الحرب في أوكرانيا وروسيا على تويتر باستخدام لغة بايثون.

تحليل معنويات حرب أوكرانيا وروسيا على تويتر باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر على Twitter بشأن حرب أوكرانيا وروسيا من Kaggle. تم جمع مجموعة البيانات هذه في البداية من Twitter ويتم تحديثها بانتظام. يمكنك تنزيل مجموعة البيانات هذه من هنا. دعنا الآن نستورد مكتبات بايثون ومجموعة البيانات اللازمة للبدء بهذه المهمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
import nltk
import re
from nltk.corpus import stopwords
import string

data = pd.read_csv)
```

```
             id  conversation_id               created_at       date     time  \
0  1.502530e+18     1.502260e+18  2022-03-12 06:03:14 UTC  3/12/2022  6:03:14
1  1.502530e+18     1.502530e+18  2022-03-12 06:03:14 UTC  3/12/2022  6:03:14
2  1.502530e+18     1.502530e+18  2022-03-12 06:03:13 UTC  3/12/2022  6:03:13
3  1.502530e+18     1.502210e+18  2022-03-12 06:03:12 UTC  3/12/2022  6:03:12
4  1.502530e+18     1.500440e+18  2022-03-12 06:03:12 UTC  3/12/2022  6:03:12

   timezone       user_id         username  \
0         0  2.019880e+07         redcelia
1         0  2.275356e+08          eee_eff
2         0  8.431317e+07      mistify_007
3         0  9.898620e+17  reallivinghuman
4         0  1.164940e+18           rpcsas
```

```
                                      name place  ... geo source user_rt_id  \
0    Johnson OutUA EUITAF💙😷 #NeverVoteTory   NaN  ... NaN    NaN        NaN
1  Wearing Masks still saves lives UAMC🏥🌹🌹    NaN  ... NaN    NaN        NaN
2                               Brian🤸   NaN  ... NaN    NaN        NaN
3                                  Basha   NaN  ... NaN    NaN        NaN
4                                 RonJon   NaN  ... NaN    NaN        NaN

  user_rt retweet_id                                           reply_to  \
0     NaN        NaN  [{'screen_name': 'RussianEmbassy', 'name': 'Ru...
1     NaN        NaN                                                 []
2     NaN        NaN                                                 []
3     NaN        NaN  [{'screen_name': 'RussianEmbassy', 'name': 'Ru...
4     NaN        NaN  [{'screen_name': 'IsraeliPM', 'name': 'Prime M...

   retweet_date  translate trans_src trans_dest
0           NaN        NaN       NaN        NaN
1           NaN        NaN       NaN        NaN
2           NaN        NaN       NaN        NaN
3           NaN        NaN       NaN        NaN
4           NaN        NaN       NaN        NaN
```

دعونا نلقي نظرة سريعة على جميع أسماء الأعمدة لمجموعة البيانات:

```
print(data.columns)
```

```
Index(['id', 'conversation_id', 'created_at', 'date', 'time', 'timezone',
       'user_id', 'username', 'name', 'place', 'tweet', 'language', 'mentions',
       'urls', 'photos', 'replies_count', 'retweets_count', 'likes_count',
       'hashtags', 'cashtags', 'link', 'retweet', 'quote_url', 'video',
       'thumbnail', 'near', 'geo', 'source', 'user_rt_id', 'user_rt',
       'retweet_id', 'reply_to', 'retweet_date', 'translate', 'trans_src',
       'trans_dest'],
      dtype='object')
```

نحتاج فقط إلى ثلاثة أعمدة لهذه المهمة (اسم المستخدم (username) والتغريدة (tweet) واللغة (language))؛ سأختار فقط هذه الأعمدة وأمضي قدمًا:

```
data = data[["username", "tweet", "language"]]
```

دعنا نلقي نظرة على ما إذا كان أي من هذه الأعمدة يحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
username    0
tweet       0
language    0
dtype: int64
```

لذلك لا يحتوي أي من الأعمدة على قيم فارغة، فلنلقِ نظرة سريعة على عدد التغريدات التي يتم نشرها بأي لغة:

```
data["language"].value_counts()
```

```
en     8812
pt      251
und     198
it      155
in      122
ru       85
hi       55
ja       52
es       40
ta       23
tr       19
ca       18
fr       16
et       16
tl       15
nl       14
de       13
pl       13
fi        9
ar        9
zh        9
sv        6
uk        6
te        6
mr        5
cs        4
el        4
gu        4
no        3
th        3
kn        3
ro        3
ur        2
or        2
eu        2
ko        2
ht        2
sl        2
bn        1
cy        1
ne        1
Name: language, dtype: int64
```

لذا فإن معظم التغريدات باللغة الإنجليزية. دعنا نجهز هذه البيانات لمهمة تحليل المشاعر. سأقوم هنا بإزالة جميع الروابط وعلامات الترقيم والرموز وأخطاء اللغة الأخرى من التغريدات:

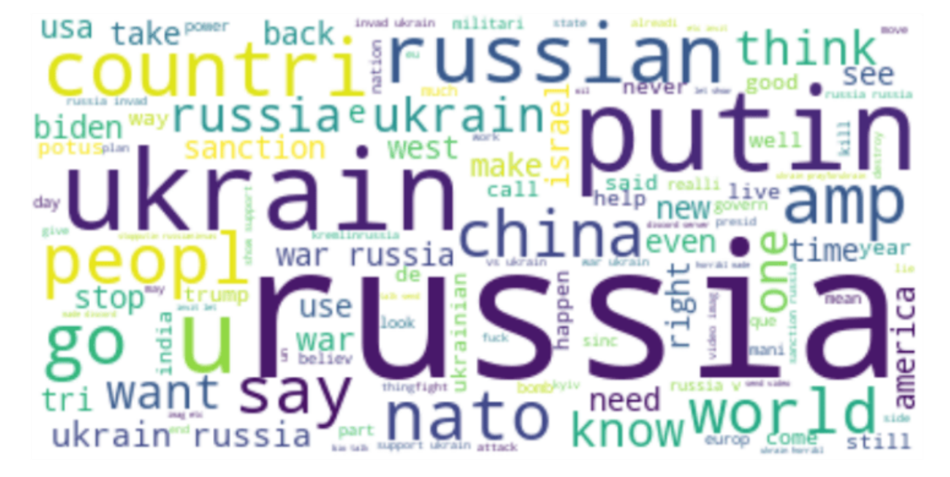
```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
```

```
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in 
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["tweet"] = data["tweet"].apply(clean)
```

دعنا الآن نلقي نظرة على سحابة الكلمات (wordcloud) في التغريدات، والتي ستظهر الكلمات الأكثر استخدامًا في التغريدات من قبل الأشخاص الذين يشاركون مشاعرهم والتحديثات حول حرب أوكرانيا وروسيا:

```
text = " ".join(i for i in data.tweet)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords, 
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



سأضيف الآن ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للتغريدات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i 
in data["tweet"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i 
in data["tweet"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i 
in data["tweet"]]
```

```
data = data[["tweet", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                               tweet  Positive  Negative  \
0  russianembassi ft mfarussia jeffdsach csdcolum...     0.077     0.284
1  kidnap without charg access lawyer putin russi...     0.000     0.000
2  much western civil everyon feel compel find cr...     0.144     0.259
3  russianembassi love place ill visit sure next ...     0.291     0.126
4  israelipm iaeaorg didnt know state israel advi...     0.000     0.000

  Neutral
0   0.639
1   1.000
2   0.596
3   0.583
4   1.000
```

دعنا الآن نلقي نظرة على الكلمات الأكثر شيوعًا التي يستخدمها الأشخاص ذوو المشاعر الإيجابية:

```
positive =' '.join([i for i in data['tweet'][data['Positive']
> data["Negative"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(positive)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



دعنا الآن نلقي نظرة على الكلمات الأكثر شيوعًا التي يستخدمها الأشخاص ذوو المشاعر السلبية:

```
negative =' '.join([i for i in data['tweet'][data['Negative']
> data["Positive"]]])
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(negative)
```

```
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



هذه هي الطريقة التي يمكنك بها تحليل مشاعر الناس بشأن حرب أوكرانيا وروسيا. آمل أن تنتهي هذه الحرب قريبًا وأن تعود الأمور إلى طبيعتها.

الملخص

هناك الكثير من التغريدات حول حرب أوكرانيا وروسيا حيث يميل الناس إلى تحديث الحقائق على الأرض، وما يشعرون به حيال ذلك، ومن يدعمون. لقد استخدمت هذه التغريداتفي مهمة تحليل المشاعر على تويتر بشأن حرب أوكرانيا وروسيا.

20) تحليل مشاعر مراجعات Flipkart باستخدام بايثون

Flipkart Reviews Sentiment Analysis using Python

Flipkart هي واحدة من أشهر الشركات الهندية. إنها منصة للتجارة الإلكترونية تتنافس مع منصات التجارة الإلكترونية الشهيرة مثل Amazon. واحدة من أكثر حالات استخدام علم البيانات شيوعًا هي مهمة تحليل المشاعر لمراجعات المنتجات المباعة على منصات التجارة الإلكترونية. لذلك، إذا كنت تريد معرفة كيفية تحليل مشاعر مراجعات Flipkart، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة Flipkart لمراجعة تحليل المشاعر باستخدام بايثون.

تحليل مشاعر مراجعات Flipkart باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها هنا لتحليل المشاعر لمراجعات Flipkart من Kaggle. لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/flipkart_reviews.csv")
print(data.head())
```

```
                                    Product_name  ... Rating
0  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
1  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
2  Lenovo Ideapad Gaming 3 Ryzen 5 Hexa Core 5600...  ...      5
3  DELL Inspiron Athlon Dual Core 3050U - (4 GB/2...  ...      5
4  DELL Inspiron Athlon Dual Core 3050U - (4 GB/2...  ...      5

[5 rows x 3 columns]
```

تحتوي مجموعة البيانات هذه على ثلاثة أعمدة فقط. دعنا نلقي نظرة على ما إذا كان أي من هذه الأعمدة يحتوي على قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

```
Product_name    0
Review          0
Rating          0
dtype: int64
```

لذلك لا تحتوي مجموعة البيانات على أي قيم فارغة. نظرًا لأن هذه هي مهمة تحليل المشاعر لمراجعاتFlipkart ، فسوف أقوم بتنظيف وإعداد العمود الذي يحتوي على المراجعات قبل التوجه إلى تحليل المشاعر:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

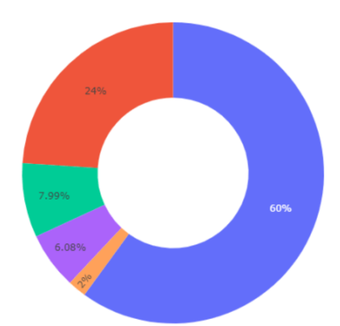
def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
  return text
data["Review"] = data["Review"].apply(clean)
```

تحليل المشاعر لمراجعاتFlipkart

يحتوي عمود التصنيف (Rating column) الخاص بالبيانات على التصنيفات التي قدمها كل مراجع. لذلك دعونا نلقي نظرة على كيفية قيام معظم الأشخاص بتقييم المنتجات التي يشترونها من Flipkart :

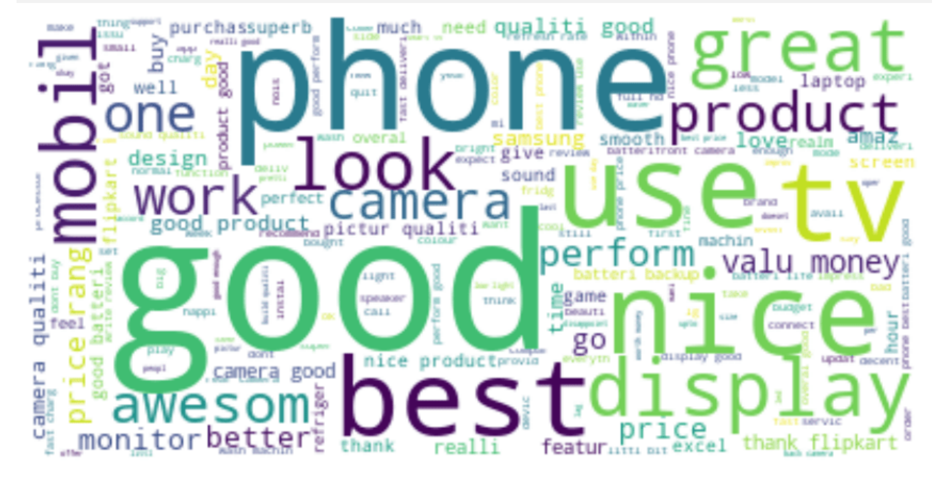
```
ratings = data["Rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

import plotly.express as px
figure = px.pie(data,
     values=quantity,
     names=numbers,hole = 0.5)
figure.show()
```



لذلك منح 60 ٪ من المراجعين 5 من أصل 5 تقييمات للمنتجات التي يشترونها من Flipkart دعونا الآن نلقي نظرة على نوع التعليقات التي يتركها الأشخاص. لهذا، سأستخدم سحابة الكلمات (word cloud) لتصوير الكلمات الأكثر استخدامًافي عمود المراجعات :

```
text = " ".join(i for i in data.Review)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
          background_color="white").generate(text(
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



سأقوم الآن بتحليل مشاعر مراجعات Flipkart عن طريق إضافة ثلاثة أعمدةفي مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) للمراجعات:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Review"]]
```

```
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Review"]]
data = data[["Review", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                               Review  ...  Neutral
0  best  great performancei got around  backup bi...  ...    0.504
1                                          good perfom  ...    0.256
2  great perform usual also game laptop issu batt...  ...    0.723
3                          wife happi best product 👌🏻😘  ...    0.488
4  light weight laptop new amaz featur batteri li...  ...    1.000

[5 rows x 4 columns]
```

الآن دعونا نرى كيف يفكر معظم المراجعين في منتجات وخدمات Flipkart:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم المراجعات محايدة. دعنا نلقي نظرة على إجمالي درجات المشاعر الإيجابية والسلبية والحيادية للعثور على نتيجة حول تقييمات Flipkart :

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  923.5529999999985
Negative:  96.77500000000013
Neutral:  1283.6880000000006
```

الملخص

لذلك، يعطي معظم الأشخاص مراجعات محايدة، وتقدم نسبة صغيرة من الأشخاص مراجعات سلبية. لذلك يمكننا القول إن الناس راضون عن منتجات وخدمات Flipkart. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر لـ Flipkart باستخدام بايثون.

21) تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون

Pfizer Vaccine Sentiment Analysis using Python

يعد Twitter أحد أكثر تطبيقات الوسائط الاجتماعية شيوعًا حيث يتمتع الأشخاص بحرية مشاركة آرائهم حول أي موضوع. هناك العديد من التغريدات المسجلة حول التوعية بلقاح فايزر (Pfizer vaccine) والتي يمكن استخدامها لتحليل مشاعر الناس حول لقاح فايزر. لذلك، إذا كنت تريد معرفة كيفية استخدام مجموعة بيانات Twitter لتحليل المشاعر، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر تجاه لقاح Pfizer باستخدام بايثون.

تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر تجاه لقاح Pfizer من Kaggle، والتي تم جمعهافي البداية من Twitter عندما كان الأشخاص يشاركون آرائهم حول لقاح Pfizer. لنبدأ مهمة تحليل المشاعر تجاه لقاح Pfizer عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/vaccination_tweets.csv")
print(data.head())
```

```
                    id            user_name  ... favorites is_retweet
0  1340539111971516416          Rachel Roh  ...         0      False
1  1338158543359250433         Albert Fong  ...         1      False
2  1337858199140118533            eliLTEU 👌  ...        0      False
3  1337855739918835717       Charles Adler  ...      2129      False
4  1337854064604966912 Citizen News Channel  ...         0      False

[5 rows x 16 columns]
```

لذا فإن مجموعة البيانات هذه كبيرة جدًا، دعنا نلقي نظرة على ما إذا كانت تحتوي على أي قيم فارغة أم لا:

```
data.isnull().sum()
```

```
id                     0
user_name              0
user_location       1630
user_description     506
user_created           0
user_followers         0
user_friends           0
user_favourites        0
user_verified          0
date                   0
text                   0
hashtags            1949
source                 1
retweets               0
favorites              0
is_retweet             0
dtype: int64
```

على الرغم من أن هذه القيم الخالية لن تؤثر على مهمة تحليل المشاعر، للحفاظ على بساطة الأمور، سأقوم بإسقاط الصفوف التي تحتوي على قيم خالية لأن مجموعة البيانات كبيرة بالفعل:

```
data = data.dropna()
```

```
                 id  user_followers  ...    retweets    favorites
count  4.749000e+03    4.749000e+03  ...  4749.000000  4749.000000
mean   1.355333e+18    5.069683e+04  ...     1.545378     9.385555
std    1.280104e+16    3.545440e+05  ...    13.395572    55.280915
min    1.337728e+18    0.000000e+00  ...     0.000000     0.000000
25%    1.344929e+18    1.740000e+02  ...     0.000000     0.000000
50%    1.352030e+18    6.480000e+02  ...     0.000000     1.000000
75%    1.364940e+18    2.728000e+03  ...     1.000000     5.000000
max    1.384788e+18    1.371493e+07  ...   678.000000  1979.000000

[8 rows x 6 columns]
```

يعد عمود النص (text column) أهم ميزةفي مجموعة البيانات هذه لأنه يحتوي على آراء مستخدمي Twitter حول لقاح فايزر. لكن يجب إعداد عمود النص لأنه يحتوي على العديد من الرموز الخاصة والأخطاء اللغوية. فيما يلي كيف يمكننا تنظيف عمود النص:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
```

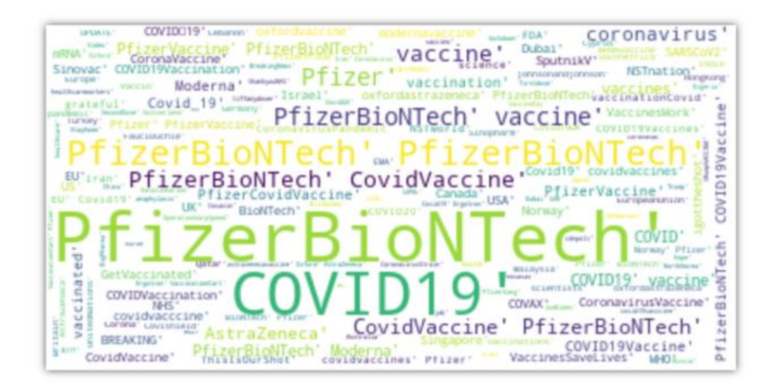
```
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["text"] = data["text"].apply(clean)
```

الآن، دعنا نلقي نظرة على سحابة الكلمات (word cloud)في عمود النص. سحابة الكلمات هي تقنية تصوير البيانات تعرض الكلمات الأكثر استخدامًا بخط كبير والكلمات الأقل استخدامًا بخط صغير. إليك كيفية تصوير سحابة الكلماتفي عمود النص:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

دعنا الآن نلقي نظرة على سحابة الكلماتفي عمود الوسوم (hashtags column)، والتي يمكن أن توضح نوع الوسوم التي كانت رائجة عندما كان الأشخاص يشاركون آرائهم حول لقاح فايزر:

```
text = " ".join(i for i in data.hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



يُظهر عمود (user_verified) في مجموعة البيانات ما إذا كان قد تم التحقق من المستخدمين الذين شاركوا آرائهم بواسطة Twitter أم لا. المستخدم الموثق على Twitter هو شخصية عامة أو شخصية مشهورة. لذلك دعونا نلقي نظرة على عدد المستخدمين الذين تم التحقق منهم والذين شاركوا آرائهم حول لقاح فايزر:

```
data["user_verified"].value_counts()
```

```
False    4169
True      580
Name: user_verified, dtype: int64
```

في الإخراج أعلاه، يُظهر False عدد المستخدمين الذين لم يتم التحقق منهم ويظهر True عدد المستخدمين الذين تم التحقق منهم. دعنا الآن ننتقل إلى مهمة تحليل المشاعر للقاح فايزر. سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                            text  ...  Neutral
0   folk said daikon past could treat cytokin stor...  ...    0.748
2   coronavirus sputnikv astrazeneca pfizerbiontec...  ...    1.000
6   bit sad claim fame success vaccin patriot comp...  ...    0.481
9   covidvaccin state start get  monday us say pak...  ...    1.000
10  death close  mark million peopl wait pfizerbio...  ...    0.698

[5 rows x 4 columns]
```

الآن دعونا نحسب شعور معظم الناس تجاه لقاح فايزر:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك كانت معظم آراء المستخدمين محايدة، دعنا نلقي نظرة على إجمالي كل نتيجة عاطفية قبل التوصل إلى أي استنتاج:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  417.81600000000003
Negative:  188.81200000000024
Neutral:  4142.3750000000055
```

مجموع الإيجابيات والسلبيات أقل بكثير من الحيادية، لذلك يمكننا القول إن مناقشة مستخدمي تويتر كانت حول الوعي بلقاح فايزر بدلاً من مشاركة فوائده أو عيوبه.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل آراء مستخدمي تويتر حول لقاح فايزر.في ختام تحليل المشاعر هذا، يمكنني القول فقط إن مناقشة مستخدمي تويتر كانت حول الوعي بلقاح فايزر بدلاً من مشاركة فوائده أو عيوبه. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر تجاه لقاح فايزر باستخدام بايثون.

22) تحليل المشاعر تجاه متحور Omicron باستخدام بايثون Omicron Sentiment Analysis using Python

قبل أيام قليلة، صنفت منظمة الصحة العالمية نوعًا جديدًا من الفيروس التاجي (coronavirus)، B.1.1.529، كمتحور مثير للقلق أطلق عليه اسم Omicron. بعد ذلك مباشرة، رأينا انتشار التغريدات حول متحور Omicron على Twitter. لذا، إذا كنت تريد معرفة كيف يمكننا تحليل مشاعر التغريدات حول متحور Omicron، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل مشاعر Omicron باستخدام بايثون.

تحليل المشاعر لمتحور Omicron باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل مشاعر Omicron من Kaggle، والتي تم جمعهافي البداية من Twitter عندما كان الأشخاص يشاركون آرائهم حول متحور Omicron. فلنبدأ مهمة تحليل المشاعر Omicron عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data = pd.read_csv("omicron.csv")
print(data.head())
```

```
                    id              user_name  ... favorites is_retweet
0  1465693385088323591                 Abaris  ...         0      False
1  1465693062999412746                 GFTs 🍌  ...         0      False
2  1465690116442279942  Herbie Finkle (Cozy)  ...         1      False
3  1465689607165591552     Electrical Review  ...         0      False
4  1465688203709464578        BingX Academy 🔑  ...         2      False

[5 rows x 16 columns]
```

مجموعة البيانات هذه كبيرة جداً، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
id                     0
user_name              0
user_location       4438
user_description    1278
user_created           0
user_followers         0
user_friends           0
user_favourites        0
user_verified          0
date                   0
text                   0
hashtags            4374
source                 0
retweets               0
favorites              0
is_retweet             0
dtype: int64
```

تحتوي مجموعة البيانات على قيم خاليةفي ثلاثة أعمدة تحتوي على بيانات نصية، وسأقوم بإزالة جميع الصفوف التي تحتوي على القيم الخالية :

```
data = data.dropna()
```

تحليل المشاعر لمتحور Omicron

يحتوي عمود النص (text column)في مجموعة البيانات على التغريدات التي قام بها الأشخاص لمشاركة آرائهم حول متحور Omicron. للمضي قدمًا، نحتاج إلى تنظيف هذا العمود وإعداده لمهمة تحليل المشاعر. إليك كيف يمكننا القيام بذلك:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
  text = str(text).lower()
  text = re.sub('\[.*?\]', '', text)
  text = re.sub('https?://\S+|www\.\S+', '', text)
  text = re.sub('<.*?>+', '', text)
  text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
  text = re.sub('\n', '', text)
  text = re.sub('\w*\d\w*', '', text)
  text = [word for word in text.split(' ') if word not in
stopword]
  text=" ".join(text)
  text = [stemmer.stem(word) for word in text.split(' ')]
  text=" ".join(text)
```

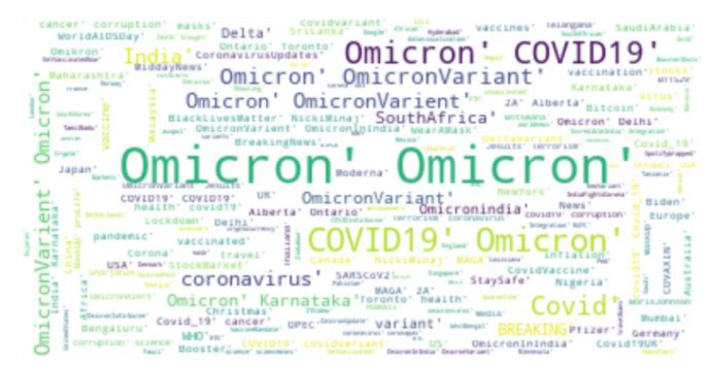
```
  return text
data["text"] = data["text"].apply(clean)
```

نظرًا لأننا قمنا بتنظيف عمود النص، فلنلقِ الآن نظرة على سحابة الكلمات (word cloud) في عمود النص لنلقي نظرة على أكبر عدد من الكلمات التي يستخدمها الأشخاص في تغريداتهم:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

دعنا الآن نلقي نظرة على سحابة الكلمات في عمود الوسوم (hashtags column) لإلقاء نظرة على أكبر عدد من علامات التصنيف التي يستخدمها الأشخاص في تغريداتهم:

```
text = " ".join(i for i in data.hashtags)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



الآن سأحسب درجات المشاعر للتغريدات حول متحور Omicron سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                                text  Positive  Negative  Neutral
0  skynew told id back omicron "odium medicum ins...      0.16     0.000    0.840
1                          someon told octob omicron      0.00     0.000    1.000
3  autom system becom increas complex effort test...      0.00     0.000    1.000
5  digitaldisrupt emerg technolog stay privat inv...      0.00     0.000    1.000
7  fatigu head bodi ach occasion sore throat coug...      0.00     0.172    0.828
```

الآن دعونا نرى كيف كان رد فعل معظم الناس حول متحور Omicron :

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
  if (a>b) and (a>c):
    print("Positive 😊 ")
  elif (b>a) and (b>c):
    print("Negative 😠 ")
  else:
    print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك كانت معظم الآراء محايدة، مما يعني أن الأشخاص كانوا يشاركون معلومات حول متغير Omicron بدلاً من مشاركة أي آراء إيجابية أو سلبية.

الملخص

إذن هذه هي الطريقة التي يمكنك بها تحليل مشاعر متحور Omicron لفيروس كورونا. إنه نوع جديد من الفيروسات التاجية التي تم تصنيفها على أنها البديل المثير للقلق من قبل منظمة الصحة العالمية. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعر Omicron باستخدام بايثون.

23) تحليل جودة المياه باستخدام بايثون Water Quality Analysis using python

يُعد الحصول على مياه الشرب المأمونة أحد الاحتياجات الأساسية لجميع البشر. من وجهة نظر قانونية، يعتبر الحصول على مياه الشرب أحد حقوق الإنسان الأساسية. تؤثر العديد من العوامل على جودة المياه (Water Quality)، كما أنها أحد مجالات البحث الرئيسيةفي التعلم الآلي. لذلك إذا كنت تريد معرفة كيفية إجراء تحليل جودة المياه (water quality analysis) باستخدام التعلم الآلي، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على تحليل جودة المياه باستخدام التعلم الآلي باستخدام بايثون.

تحليل جودة المياه

يُعد تحليل جودة المياه أحد المجالات الرئيسية للبحثفي التعلم الآلي. يُعرف أيضًا باسم تحليل قابلية المياه للشرب (water potability analysis) لأن مهمتنا هنا هي فهم جميع العوامل التي تؤثر على قابلية المياه للشرب وتدريب نموذج التعلم الآلي الذي يمكنه تصنيف ما إذا كانت عينة مياه معينة آمنة أو غير صالحة للاستهلاك.

بالنسبة لمهمة تحليل جودة المياه، سأستخدم مجموعة بيانات Kaggle التي تحتوي على بيانات حول جميع العوامل الرئيسية التي تؤثر على قابلية المياه للشرب. جميع العوامل التي تؤثر على جودة المياه مهمة للغاية، لذلك نحتاج إلى استكشاف كل ميزة من ميزات مجموعة البيانات هذه بإيجاز قبل تدريب نموذج التعلم الآلي للتنبؤ بما إذا كانت عينة المياه آمنة أو غير مناسبة للاستهلاك. يمكنك تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل جودة المياه من هنا.

تحليل جودة المياه باستخدام لغة بايثون

سأبدأ مهمة تحليل جودة المياه عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np

data = pd.read_csv("water_potability.csv")
data.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

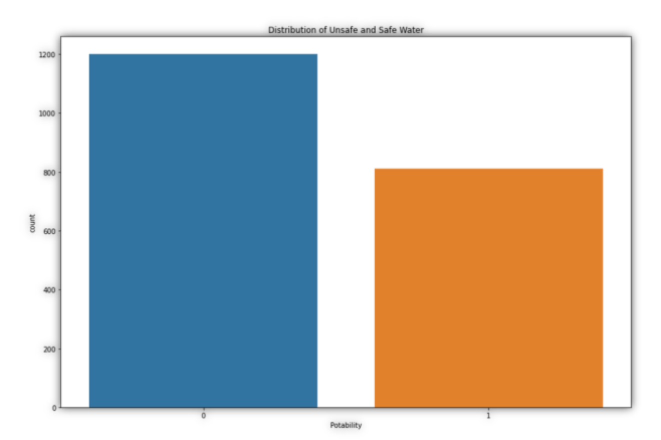
يمكنني رؤية القيم الخاليةفي المعاينة الأولى لمجموعة البيانات هذه نفسها، لذلك قبل المضي قدمًا، دعنا نزيل جميع الصفوف التي تحتوي على قيم فارغة:

```
data = data.dropna()
data.isnull().sum()
```

```
ph                 0
Hardness           0
Solids             0
Chloramines        0
Sulfate            0
Conductivity       0
Organic_carbon     0
Trihalomethanes    0
Turbidity          0
Potability         0
dtype: int64
```

عمود القابلية للشرب (Potability column) لمجموعة البيانات هذه هو العمود الذي نحتاج إلى توقعه لأنه يحتوي على القيمتين 0 و 1 التي تشير إلى ما إذا كانت المياه صالحة للشرب (1) أو غير صالحة (0) للشرب. لذلك دعونا نرى توزيع 0 و 1في عمود "Potability ":

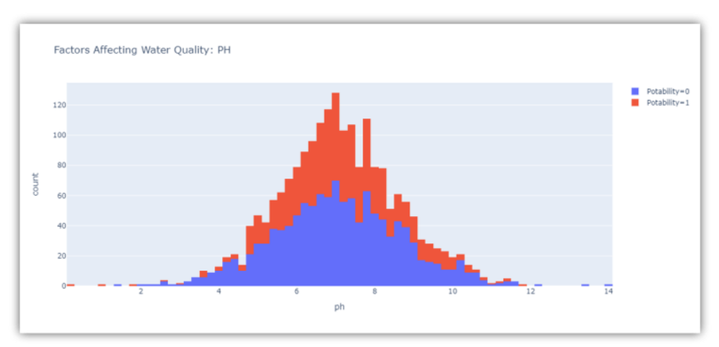
```
plt.figure(figsize=(15, 10))
sns.countplot(data.Potability)
plt.title("Distribution of Unsafe and Safe Water")
plt.show()
```



لذلك هذا شيء يجب أن تلاحظه أن مجموعة البيانات هذه غير متوازنة (not balanced) لأن عينات الأصفار أكثر من 1 ثانية.

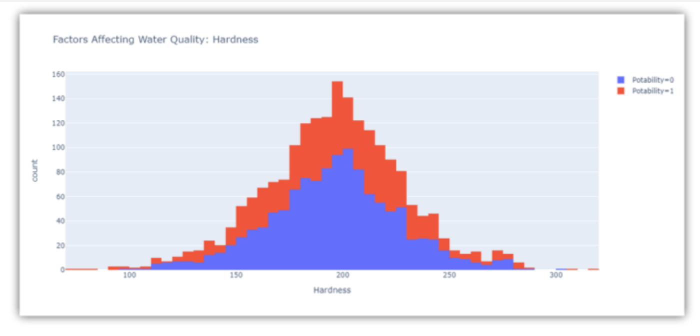
كما ذكرنا أعلاه، لا توجد عوامل لا يمكننا تجاهلها والتي تؤثر على جودة المياه، لذلك دعونا نستكشف جميع الأعمدة واحدة تلو الأخرى. لنبدأ بإلقاء نظرة على عمود الاس الهيدروجيني (ph column):

```
import plotly.express as px
data = data
figure = px.histogram(data, x = "ph",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
PH")
figure.show()
```



يمثل عمود الأس الهيدروجيني قيمة الأس الهيدروجيني للماء وهو عامل مهمفي تقييم التوازن الحمضي القاعدي للماء. يجب أن تكون قيمة الرقم الهيدروجيني لمياه الشرب بين 6.5 و8.5. دعونا الآن نلقي نظرة على العامل الثاني الذي يؤثر على جودة المياهفي مجموعة البيانات:

```
figure = px.histogram(data, x = "Hardness",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Hardness")
figure.show()
```



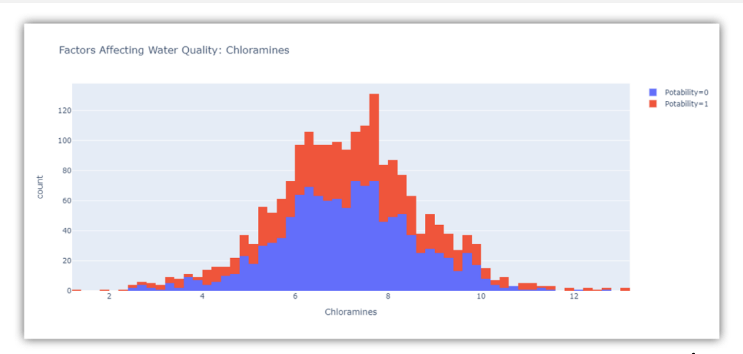
يوضح الشكل أعلاه توزيع عسرة hardness الماءفي مجموعة البيانات. تعتمد عسرة الماء عادة على مصدره، لكن الماء الذي تصل قوته إلى 120ـ200 ملليغرام صالح للشرب. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Solids",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Solids")
figure.show()
```



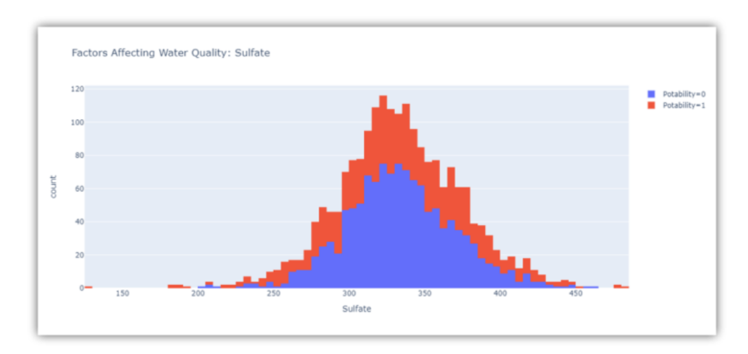
يمثل الشكل أعلاه توزيع إجمالي المواد الصلبة الذائبةفي الماءفي مجموعة البيانات. تسمى جميع المعادن العضوية وغير العضوية الموجودةفي الماء بالمواد الصلبة الذائبة. الماء الذي يحتوي على عدد كبير جداً من المواد الصلبة الذائبة شديد التمعدن. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Chloramines",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Chloramines")
figure.show()
```



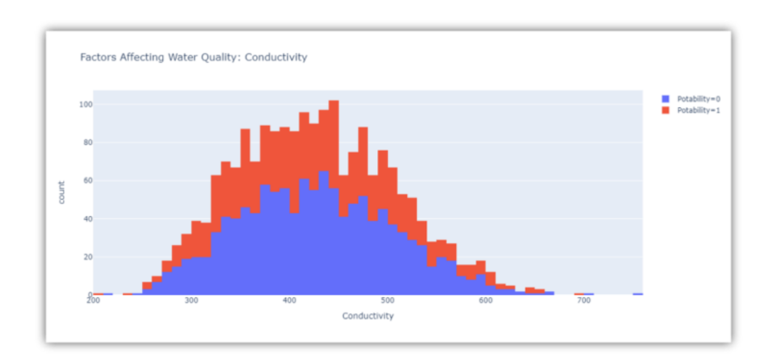
يمثل الشكل أعلاه توزيع الكلورامين (chloramine)في الماءفي مجموعة البيانات. الكلورامين والكلور من المطهرات المستخدمةفي أنظمة المياه العامة. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة المياه:

```
figure = px.histogram(data, x = "Sulfate",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Sulfate")
figure.show()
```



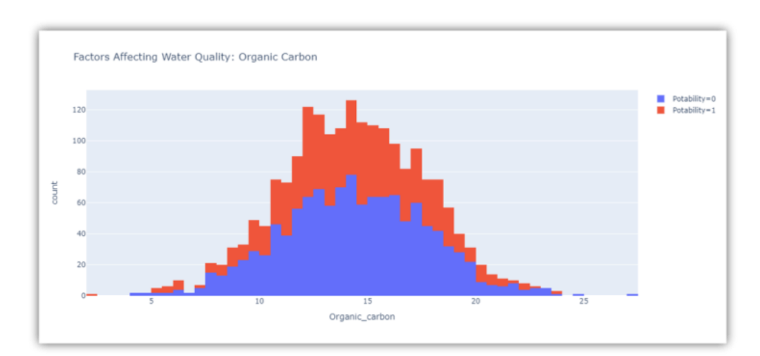
يوضح الشكل أعلاه توزيع الكبريتات (sulfate)في الماءفي مجموعة البيانات. إنها مواد موجودة بشكل طبيعيفي المعادن والتربة والصخور. الماء الذي يحتوي على أقل من 500 ملليغرام من الكبريتات آمن للشرب. الآن دعونا نرى العامل التالي:

```
figure = px.histogram(data, x = "Conductivity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Conductivity")
figure.show()
```



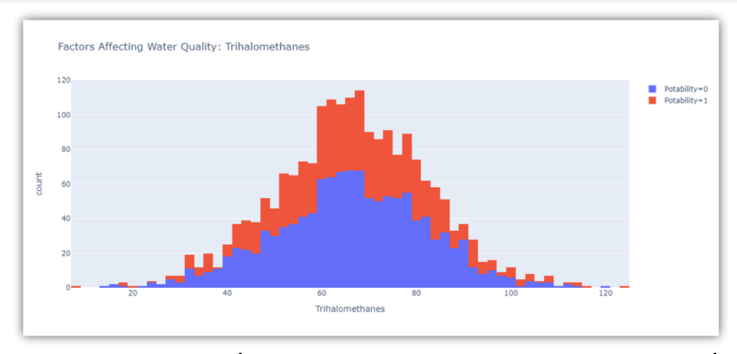
يمثل الشكل أعلاه توزيع موصلية المياه (water conductivity)في مجموعة البيانات. الماء موصل جيد للكهرباء، لكن أنقى أشكال الماء ليس موصلًا جيدًا للكهرباء. المياه ذات التوصيل الكهربائي أقل من 500 صالحة للشرب. الآن دعونا نرى العامل التالي:

```
figure = px.histogram(data, x = "Organic_carbon",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Organic Carbon")
figure.show()
```



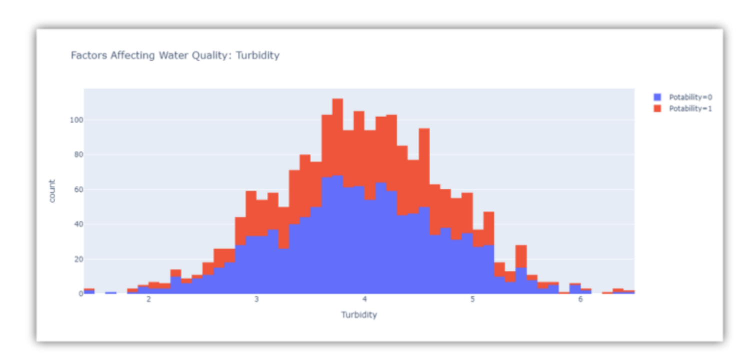
يمثل الشكل أعلاه توزيع الكربون العضوي (organic carbon)في الماءفي مجموعة البيانات. يأتي الكربون العضوي من انهيار المواد العضوية الطبيعية والمصادر الاصطناعية. تعتبر المياه التي تحتوي على أقل من 25 ملليجرام من الكربون العضوي آمنة للشرب. دعنا الآن نلقي نظرة على العامل التالي الذي يؤثر على جودة مياه الشرب:

```
figure = px.histogram(data, x = "Trihalomethanes",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Trihalomethanes")
figure.show()
```



يمثل الشكل أعلاه توزيع ثلاثي الميثان (trihalomethanes) أو (THM)في الماءفي مجموعة البيانات. THMs هي مواد كيميائية موجودةفي المياه المعالجة بالكلور. تعتبر المياه التي تحتوي على أقل من 80 ملليجرام من THMs آمنة للشرب. دعنا الآن نلقي نظرة على العامل التاليفي مجموعة البيانات الذي يؤثر على جودة مياه الشرب:

```
figure = px.histogram(data, x = "Turbidity",
                      color = "Potability",
                      title= "Factors Affecting Water Quality:
Turbidity")
figure.show()
```



يمثل الشكل أعلاه توزيع العكارة (turbidity)في الماء. تعتمد عكارة الماء على عدد المواد الصلبة الموجودةفي المعلق. تعتبر المياه ذات العكارة أقل من 5 ملليغرام صالحة للشرب.

نموذج التنبؤ بجودة المياه باستخدام لغة بايثون

في القسم أعلاه، استكشفنا جميع الميزات التي تؤثر على جودة المياه. الآن، الخطوة التالية هي تدريب نموذج التعلم الآلي لمهمة تحليل جودة المياه باستخدام بايثون. لهذه المهمة، سأستخدم مكتبة PyCaretفي بايثون. إذا لم تكن قد استخدمت هذه المكتبة من قبل، فيمكنك تثبيتها بسهولة على نظامك باستخدام الأمر pip:

```
pip install pycaret
```

قبل تدريب نموذج التعلم الآلي، دعنا نلقي نظرة على الارتباط (correlation) بين جميع الميزات فيما يتعلق بعمود القابلية للشرب (Potability column)في مجموعة البيانات:

```
correlation = data.corr()
correlation["ph"].sort_values(ascending=False)
```

```
ph                 1.000000
Hardness           0.108948
Organic_carbon     0.028375
Trihalomethanes    0.018278
Potability         0.014530
Conductivity       0.014128
Sulfate            0.010524
Chloramines       -0.024768
Turbidity         -0.035849
Solids            -0.087615
Name: ph, dtype: float64
```

الآن فيما يلي كيف يمكنك معرفة أي خوارزمية تعلم الآلة هي الأفضل لمجموعة البيانات هذه باستخدام مكتبة PyCaretفي بايثون:

```
from pycaret.classification import*
clf = setup(data, target = "Potability", silent = True,
session_id = 786)
compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.6830	0.7005	0.4197	0.6744	0.5133	0.2976	0.3182	0.724
qda	Quadratic Discriminant Analysis	0.6823	0.7192	0.3985	0.6883	0.5013	0.2917	0.3174	0.022
et	Extra Trees Classifier	0.6816	0.6941	0.3861	0.6858	0.4916	0.2863	0.3123	0.557
lightgbm	Light Gradient Boosting Machine	0.6652	0.6916	0.4762	0.6078	0.5324	0.2781	0.2840	0.172
gbc	Gradient Boosting Classifier	0.6602	0.6738	0.3718	0.6306	0.4667	0.2419	0.2603	0.339
nb	Naive Bayes	0.6184	0.6078	0.2478	0.5545	0.3412	0.1261	0.1462	0.019
dt	Decision Tree Classifier	0.6034	0.5895	0.5186	0.5049	0.5097	0.1775	0.1784	0.027
lr	Logistic Regression	0.5984	0.5199	0.0071	0.1900	0.0134	0.0028	0.0127	0.355
ridge	Ridge Classifier	0.5984	0.0000	0.0089	0.1583	0.0168	0.0035	0.0056	0.021
lda	Linear Discriminant Analysis	0.5977	0.4903	0.0089	0.1500	0.0167	0.0021	0.0024	0.022
ada	Ada Boost Classifier	0.5956	0.5671	0.2919	0.4896	0.3644	0.0972	0.1034	0.173
knn	K Neighbors Classifier	0.5743	0.5423	0.3644	0.4642	0.4070	0.0826	0.0846	0.121
svm	SVM - Linear Kernel	0.5194	0.0000	0.3982	0.1604	0.2287	-0.0014	-0.0104	0.027

وفقًا للنتيجة أعلاه، فإن خوارزمية الغابة العشوائية random forrest(rf) هي الأفضل لتدريب نموذج التعلم الآلي لمهمة تحليل جودة المياه. لذلك دعونا ندرب النموذج ونفحص تنبؤاته:

```
model = create_model("rf")
predict = predict_model(model, data=data)
predict.head()
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability	Label	Score
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0	0	0.87
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0	0	0.91
5	5.584087	188.313324	28748.687739	7.544869	326.678363	280.467916	8.399735	54.917862	2.559708	0	0	0.83
6	10.223862	248.071735	28749.716544	7.513408	393.663396	283.651634	13.789695	84.603556	2.672989	0	0	0.89
7	8.635849	203.361523	13672.091764	4.563009	303.309771	474.607645	12.363817	62.798309	4.401425	0	0	0.94

النتائج المذكورة أعلاه تبدو مرضية. آمل أن تكون قد أحببت مشروع التعلم الآلي هذا حول تحليل جودة المياه باستخدام بايثون.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل جودة المياه وتدريب نموذج التعلم الآلي لتصنيف المياه الآمنة وغير الآمنة للشرب. يُعد الحصول على مياه الشرب المأمونة أحد الاحتياجات الأساسية لجميع البشر. من وجهة نظر قانونية، يعتبر الحصول على مياه الشرب أحد حقوق الإنسان الأساسية. تؤثر العديد من العوامل على جودة المياه، كما أنها أحد مجالات البحث الرئيسيةفي التعلم الآلي.

24) تحليل المشاعر على Twitter باستخدام بايثون Twitter Sentiment Analysis using Python

Twitter هو أحد منصات التواصل الاجتماعي حيث يتمتع الأشخاص بحرية مشاركة آرائهم حول أي موضوع. نرى أحيانًا مناقشة قوية على Twitter حول رأي شخص ما تؤدي أحيانًا إلى مجموعة من التغريدات السلبية. مع وضع ذلك في الاعتبار، إذا كنت تريد معرفة كيفية إجراء تحليل المشاعر (sentiment analysis) على Twitter، فهذه المقالة مناسبة لك. في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على Twitter

تحليل المشاعر مهمة معالجة اللغة الطبيعية (natural language processing). يجب على جميع منصات وسائل التواصل الاجتماعي مراقبة مشاعر المشاركين في المناقشة. نرى في الغالب آراء سلبية على تويتر عندما تكون المناقشة سياسية. لذلك، يجب أن تستمر كل منصة في تحليل المشاعر للعثور على نوع الأشخاص الذين ينشرون الكراهية والسلبية على نظامهم الأساسي.

بالنسبة لمهمة تحليل المشاعر على Twitter، قمت بجمع مجموعة بيانات من Kaggle تحتوي على تغريدات حول مناقشة طويلة داخل مجموعة من المستخدمين. مهمتنا هنا هي تحديد عدد التغريدات السلبية والإيجابية حتى نتمكن من إعطاء نتيجة. لذلك، في القسم أدناه، سأقدم لك مهمة تحليل المشاعر على Twitter باستخدام بايثون.

تحليل المشاعر على Twitter باستخدام بايثون

لنبدأ مهمة تحليل المشاعر على Twitter من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import re
import nltk
import nltk

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/twitter.csv")
print(data.head())
```

```
   Unnamed: 0  count  hate_speech  offensive_language  neither  class  \
0           0      3            0                   0        3      2
1           1      3            0                   3        0      1
2           2      3            0                   3        0      1
3           3      3            0                   2        1      1
4           4      6            0                   6        0      1

                                               tweet
0  !!! RT @mayasolovely: As a woman you shouldn't...
1  !!!!! RT @mleew17: boy dats cold...tyga dwn ba...
2  !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...
3  !!!!!!!!! RT @C_G_Anderson: @viva_based she lo...
4  !!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...
```

يحتوي عمود التغريدة(tweet column)في مجموعة البيانات أعلاه على التغريدات التي نحتاج إلى استخدامها لتحليل مشاعر المشاركين في المناقشة. ولكن للمضي قدمًا، يتعين علينا تنظيف الكثير من الأخطاء والرموز الخاصة الأخرى لأن هذه التغريدات تحتوي على الكثير من الأخطاء اللغوية. إذن إليك كيف يمكننا تنظيف عمود التغريدة:

```
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["tweet"] = data["tweet"].apply(clean)
```

الآن، الخطوة التالية هي حساب درجات المشاعر (sentiment scores) لهذه التغريدات وتعيين تسمية للتغريدات على أنها إيجابية (Posetive) أو سلبية (Negative) أو محايدة (Neutral). إليك كيفية حساب درجات المشاعرفي التغريدات:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["tweet"]]
```

```
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["tweet"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["tweet"]]
```

الآن سأختار فقط الأعمدة من هذه البيانات التي نحتاجها لبقية مهمة تحليل المشاعر على Twitter:

```
data = data[["tweet", "Positive",
             "Negative", "Neutral"]]
print(data.head())
```

```
                                               tweet  Positive  Negative  \
0  rt mayasolov woman shouldnt complain clean ho...     0.147     0.157
1  rt  boy dat coldtyga dwn bad cuffin dat hoe  ...     0.000     0.280
2  rt urkindofbrand dawg rt  ever fuck bitch sta...     0.000     0.577
3            rt cganderson vivabas look like tranni     0.333     0.000
4  rt shenikarobert shit hear might true might f...     0.154     0.407

   Neutral
0    0.696
1    0.720
2    0.423
3    0.667
4    0.440
```

دعنا الآن نلقي نظرة على التصنيف الأكثر شيوعًا المخصص للتغريدات وفقًا لدرجات المشاعر:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم التغريدات محايدة (neutral)، ما يعني أنها ليست إيجابية ولا سلبية. الآن دعنا نلقي نظرة على إجمالي درجات المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  2880.086000000009
Negative:  7201.020999999922
Neutral:  14696.887999999733
```

مجموع التغريدات المحايدة أعلى بكثير من السلبية والإيجابية، لكن من بين جميع التغريدات السلبية أكبر من التغريدات الإيجابية، لذلك يمكننا القول إن معظم الآراء سلبية.

الملخص

هذه هي الطريقة التي يمكنك بها أداء مهمة تحليل المشاعر على Twitter باستخدام لغة برمجة بايثون. تحليل المشاعر مهمة معالجة اللغة الطبيعية. تحتاج جميع منصات وسائل التواصل الاجتماعي إلى التحقق من مشاعر الأشخاص المشاركين في المناقشة. آمل أن تكون قد أحببت هذا المقال على تحليل المشاعر على Twitter باستخدام بايثون.

25) تحليل مشاعر لعبة الحبار باستخدام بايثون Squid Game Sentiment Analysis using Python

تعد لعبة الحبار (squid game) حاليًا واحدة من أكثر العروض شيوعًا على Netflix. من الشائع جداً أن الأشخاص الذين لم يشاهدوا أي سلسلة ويب من قبل يشاهدونها أيضًا. أحد أسباب ذلك هو آراء وآراء المشاهدين على وسائل التواصل الاجتماعي. لذلك إذاكنت تريد معرفة كيفية تحليل مشاعر الناس حول لعبة الحبار، فهذه المقالة مناسبة لك.في هذه المقالة، سوف آخذك خلال مهمة تحليل مشاعر لعبة الحبار باستخدام بايثون.

تحليل المشاعر لعبة الحبار باستخدام بايثون

يتم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل مشاعر لعبة الحبار من Kaggle، والتي تم جمعهافي البداية من Twitter بينما كان الأشخاص يشاركون بنشاط آرائهم حول لعبة الحبار. لنبدأ مهمة تحليل مشاعر لعبة الحبار عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

data = pd.read_csv("squid_game.csv")
print(data.head())
```

```
                  user_name  user_location  ...              source is_retweet
0   the _ûndër-ratèd niggáh✊🏾            NaN  ...  Twitter for Android      False
1  Best uncle on planet earth            NaN  ...  Twitter for Android      False
2                      marcie            NaN  ...      Twitter Web App      False
3                   YoMo.Mdp   Any pronouns  ...      Twitter Web App      False
4             Laura Reactions         France  ...      Twitter Web App      False

[5 rows x 12 columns]
```

في الانطباعات الأولى لمجموعة البيانات هذه، لاحظت قيمًا خاليةفي عمود (user_location) يبدو أنها لا تؤثر على مهمة تحليل المشاعر. لذلك سوف أسقط هذا العمود:

```
data = data.drop(columns="user_location", axis=1)
```

دعنا الآن نلقي نظرة على ما إذاكانت الأعمدة الأخرى تحتوي على أي قيم فارغة أم لا:

```
print(data.isnull().sum())
```

```
user_name            4
user_description  5211
user_created         0
user_followers       0
user_friends         0
user_favourites      0
user_verified        0
date                 0
text                 0
source               0
is_retweet           0
dtype: int64
```

يحتوي عمود (user_description) أيضًا على قيم خالية، والتي لن تؤثر أيضًا على مهمة تحليل المشاعر. لذلك سأحذف هذا العمود أيضًا:

```
data = data.drop(columns="user_description", axis=1)
data = data.dropna)
```

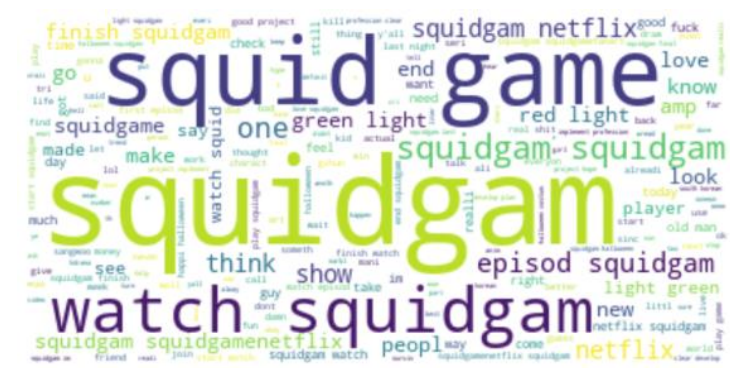
يحتوي عمود النص (text)في مجموعة البيانات على آراء مستخدمي تويتر حول لعبة الحبار، فهذه آراء لوسائل التواصل الاجتماعي، لذا يجب تحضير هذا العمود قبل أي تحليل. لذلك دعونا نجهز هذا العمود لمهمة تحليل المشاعر:

```
import nltk
import re
nltk.download('stopwords')
stemmer = nltk.SnowballStemmer("english")
from nltk.corpus import stopwords
import string
stopword=set(stopwords.words('english'))

def clean(text):
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '',
text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = [word for word in text.split(' ') if word not in
stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text
data["text"] = data["text"].apply(clean)
```

دعنا الآن نلقي نظرة على الكلمات الأكثر استخدامًافي آراء لعبة الحبار باستخدام سحابة الكلمات (word cloud). سحابة الكلمات هي أداة تصوير البيانات تعرض الكلمات الأكثر استخدامًا بحجم أكبر. إليك كيفية تصوير سحابة الكلماتفي عمود النص:

```
text = " ".join(i for i in data.text)
stopwords = set(STOPWORDS)
wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
plt.figure( figsize=(15,10))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



الآن دعنا ننتقل إلى مهمة تحليل المشاعرفي لعبة الحبار. سأضيف هنا ثلاثة أعمدة أخرى في مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر (sentiment scores) في عمود النص:

```
nltk.download('vader_lexicon')
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["text"]]
data = data[["text", "Positive", "Negative", "Neutral"]]
print(data.head())
```

```
                                               text  Positive  Negative  Neutral
0  life hit time poverti strike yougong yoo  let ...     0.173     0.108    0.719
1                      marbl episod squidgam  ruin 😭😭😭     0.000     0.487    0.513
2                                     squidgam time     0.000     0.000    1.000
3  blood  slideim join squidgam thing im alreadi ...     0.142     0.277    0.581
4  two first game player kill mask guy  bloodi ni...     0.000     0.461    0.539
```

دعنا الآن نحسب كيف يفكر معظم الناس في لعبة الحبار:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
```

```
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذا فإن معظم آراء المستخدمين حيادية، فلنلقِ الآن نظرة على إجمالي كل درجة المشاعر قبل التوصل إلى أي استنتاج:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  10604.55899999976
Negative:  5171.334000000031
Neutral:  64233.11800000302
```

إجمالي السلبيات أقل بكثير من الإيجابية، لذلك يمكننا القول إن معظم الآراء حول لعبة الحبار إيجابية.

الملخص

تعد لعبة الحبار حاليًا واحدة من أكثر العروض شيوعًا على Netflix. أحد أسباب ذلك هو آراء وآراء المشاهدين على وسائل التواصل الاجتماعي. آمل أن تكون قد أحببت هذا المقال حول تحليل المشاعرفي لعبة الحبار باستخدام بايثون.

26) تحليل تصنيف الفيلم باستخدام بايثون Movie Rating Analysis using Python

نشاهد جميعًا الأفلام للترفيه، والبعض منا لا يقيمها أبدًا، بينما يقوم بعض المشاهدين دائمًا بتقييم كل فيلم يشاهدونه. يساعد هذا النوع من المشاهدين في تصنيف الأفلام للأشخاص الذين يراجعون مراجعات الفيلم قبل مشاهدة أي فيلم للتأكد من أنهم على وشك مشاهدة فيلم جيد. لذلك، إذا كنت جديداً في علم البيانات وترغب في معرفة كيفية تحليل تقييمات الأفلام باستخدام لغة برمجة بايثون، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل تصنيف الأفلام (Movie Rating Analysis) باستخدام بايثون.

تحليل تصنيف الفيلم باستخدام بايثون

يساعد تحليل التصنيف الذي قدمه مشاهدو الفيلم العديد من الأشخاص في تحديد ما إذا كانوا سيشاهدون هذا الفيلم أم لا. لذلك، بالنسبة لمهمة تحليل تصنيف الفيلم، تحتاج أولاً إلى مجموعة بيانات تحتوي على بيانات حول التصنيفات التي قدمها كل عارض. لهذه المهمة، قمت بجمع مجموعة بيانات من Kaggle تحتوي على ملفين:

1. يحتوي ملف واحد على بيانات حول معرف الفيلم (movie Id) وعنوانه (title) ونوعه (genre).

2. ويحتوي الملف الآخر على معرف المستخدم (user id) ومعرف الفيلم (movie id) والتقييمات (ratings) التي قدمها المستخدم والطابع الزمني للتصنيفات.

يمكنك تنزيل مجموعتي البيانات هاتين من هنا.

لنبدأ الآن بمهمة تحليل تصنيف الأفلام عن طريق استيراد مكتبات بايثون ومجموعات البيانات الضرورية:

```
import numpy as np
import pandas as pd
movies = pd.read_csv("movies.dat", delimiter='::')
print(movies.head())
```

```
0      10               La sortie des usines Lumière (1895)     Documentary|Short
1      12                    The Arrival of a Train (1896)     Documentary|Short
2      25  The Oxford and Cambridge University Boat Race ...                NaN
3      91                       Le manoir du diable (1896)          Short|Horror
4     131                         Une nuit terrible (1896)   Short|Comedy|Horror
```

في الكود أعلاه، قمت فقط باستيراد مجموعة بيانات الأفلام التي لا تحتوي على أي أسماء أعمدة، لذلك دعونا نحدد أسماء الأعمدة:

```
movies.columns = ["ID", "Title", "Genre"]
print(movies.head())
```

```
    ID                                              Title                 Genre
0   10                     La sortie des usines Lumière (1895)   Documentary|Short
1   12                        The Arrival of a Train (1896)   Documentary|Short
2   25  The Oxford and Cambridge University Boat Race ...                   NaN
3   91                           Le manoir du diable (1896)        Short|Horror
4  131                            Une nuit terrible (1896)  Short|Comedy|Horror
```

الآن دعنا نستورد مجموعة بيانات التصنيفات (ratings dataset):

```
ratings = pd.read_csv("ratings.dat", delimiter='::')
print(ratings.head())
```

```
   1  0114508  8  1381006850
0  2   499549  9  1376753198
1  2  1305591  8  1376742507
2  2  1428538  1  1371307089
3  3    75314  1  1595468524
4  3   102926  9  1590148016
```

لا تحتوي مجموعة بيانات التصنيف أيضًا على أي أسماء أعمدة، لذلك دعونا نحدد أسماء الأعمدة لهذه البيانات أيضًا:

```
ratings.columns = ["User", "ID", "Ratings", "Timestamp"]
print(ratings.head())
```

```
   User       ID  Ratings   Timestamp
0     2   499549        9  1376753198
1     2  1305591        8  1376742507
2     2  1428538        1  1371307089
3     3    75314        1  1595468524
4     3   102926        9  1590148016
```

سأقوم الآن بدمج مجموعتي البيانات هاتين في واحدة، تحتوي مجموعتي البيانات هاتين على عمود مشترك كمعرف، والذي يحتوي على معرف الفيلم (movie ID)، لذلك يمكننا استخدام هذا العمود كعمود مشترك لدمج مجموعتي البيانات:

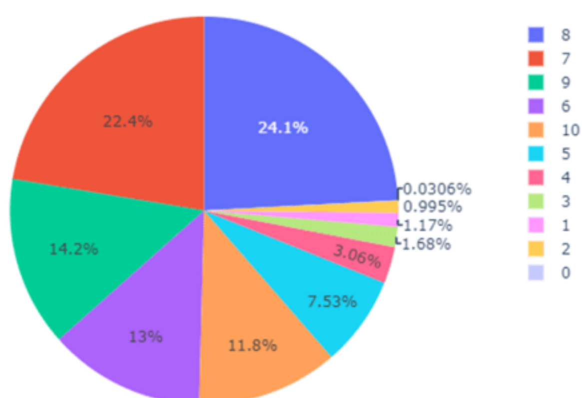
```
data = pd.merge(movies, ratings, on=["ID", "ID"])
print(data.head())
```

```
   ID                                              Title  ... Ratings   Timestamp
0  10                     La sortie des usines Lumière (1895)  ...      10  1412878553
1  12                        The Arrival of a Train (1896)  ...      10  1439248579
2  25  The Oxford and Cambridge University Boat Race ...  ...       8  1488189899
3  91                           Le manoir du diable (1896)  ...       6  1385233195
4  91                           Le manoir du diable (1896)  ...       5  1532347349

[5 rows x 6 columns]
```

نظرًا لأنها مهمة على مستوى المبتدئين، سأقوم أولاً بإلقاء نظرة على توزيع تصنيفات جميع الأفلام التي قدمها المشاهدون:

```
ratings = data["Ratings"].value_counts()
numbers = ratings.index
quantity = ratings.values
import plotly.express as px
fig = px.pie(data, values=quantity, names=numbers)
fig.show()
```



لذلك، وفقًا لمخطط الدائرة المجزأة (pie chart) أعلاه، تم تصنيف معظم الأفلام 8 من قبل المستخدمين. من الشكل أعلاه، يمكن القول إن معظم الأفلام تم تصنيفها بشكل إيجابي.

نظرًا لأن الرقم 10 هو أعلى تصنيف يمكن للمشاهد تقديمه، فلنلقِ نظرة على أفضل 10 أفلام حصلت على 10 تقييمات من قبل المشاهدين:

```
data2 = data.query("Ratings == 10")
print(data2["Title"].value_counts().head(10))
```

```
Joker (2019)                        1479
Interstellar (2014)                 1382
1917 (2019)                          819
Avengers: Endgame (2019)             808
The Shawshank Redemption (1994)      699
Gravity (2013)                       653
The Wolf of Wall Street (2013)       581
Hacksaw Ridge (2016)                 570
Avengers: Infinity War (2018)        534
La La Land (2016)                    510
Name: Title, dtype: int64
```

لذلك، وفقًا لمجموعة البيانات هذه، حصل (Joker2019) على أعلى عدد 10 تقييمات من المشاهدين. هذه هي الطريقة التي يمكنك بها تحليل تقييمات الأفلام باستخدام بايثون كمبتدئ في علم البيانات.

الملخص

هذه هي الطريقة التي يمكنك بها إجراء تحليل تصنيف الفيلم باستخدام لغة برمجة بايثون كمبتدئ في علوم البيانات. يساعد تحليل التصنيفات التي قدمها مشاهدو الفيلم العديد من الأشخاص في تحديد ما إذا كانوا سيشاهدون هذا الفيلم أم لا. آمل أن تكون قد أحببت هذه المقالة حول تحليل تصنيف الأفلام باستخدام بايثون.

27) تحليل المليارديرات مع بايثون Billionaires Analysis with Python

يقول عدد المليارديرات (Billionaires)في بلد ما الكثير عن بيئة الأعمال ومعدل نجاح بدء التشغيل والعديد من الميزات الاقتصادية الأخرى للبلد. لذلك إذا كنت تريد معرفة المزيد حول كيفية إيجاد علاقات بين المليارديرات حول العالم، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المليارديرات باستخدام بايثون.

تحليل المليارديرات مع بايثون

مجموعة البيانات التي أستخدمها لتحليل البيانات حول المليارديرات حول العالم برعاية مجلة Forbes ويتم تنزيلها من Kaggle. تحتوي مجموعة البيانات على معلومات حول المليارديرات العالميين في عام 2021، بمافي ذلك:

1. الأسماء (Names).
2. صافي القيمة (Net Worth).
3. الدولة (Country).
4. المصدر (Source).
5. المرتبة (Rank).
6. العمر (Age).
7. الصناعة (Industry).

فلنبدأ بمهمة تحليل المليارديرات عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt


data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/Billionaire.csv")
print(data.head())
```

```
                     Name NetWorth        Country          Source  Rank   Age          Industry
0              Jeff Bezos   $177 B  United States          Amazon     1  57.0        Technology
1               Elon Musk   $151 B  United States  Tesla, SpaceX     2  49.0        Automotive
2  Bernard Arnault & family   $150 B         France            LVMH     3  72.0  Fashion & Retail
3              Bill Gates   $124 B  United States       Microsoft     4  65.0        Technology
4         Mark Zuckerberg    $97 B  United States        Facebook     5  36.0        Technology
```

قبل المضي قدمًا، دعنا نرى ما إذاكانت مجموعة البيانات هذه تحتوي على قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

```
Name         0
NetWorth     0
Country      0
Source       0
Rank         0
Age         79
Industry     0
dtype: int64
```

إذن، تحتوي مجموعة البيانات هذه على 79 قيمة مفقودةفي عمود العمر (Age column)، فلنزيل هذه الصفوف:

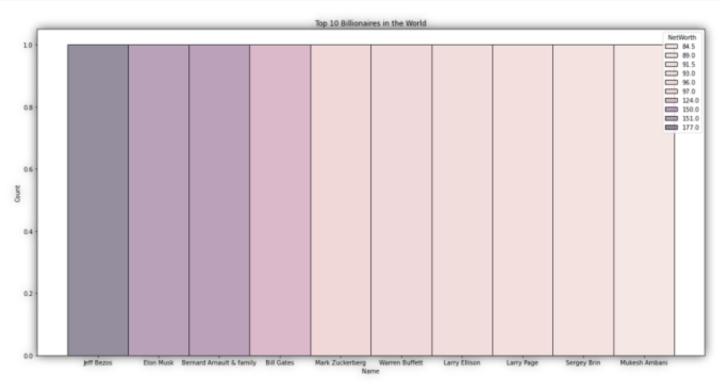
```
data = data.dropna()
```

يحتوي عمود صافي الثروة (NetWorth)في مجموعة البيانات هذه على علامة $في بداية صافي ثروة المليارديرات وBفي النهاية. لذلك نحتاج إلى إزالة هذه العلامات وتحويل عمود صافي الثروة إلى قيمة عائمة (float):

```
data["NetWorth"] = data["NetWorth"].str.strip("$")
data["NetWorth"] = data["NetWorth"].str.strip("B")
data["NetWorth"] = data["NetWorth"].astype(float)
```

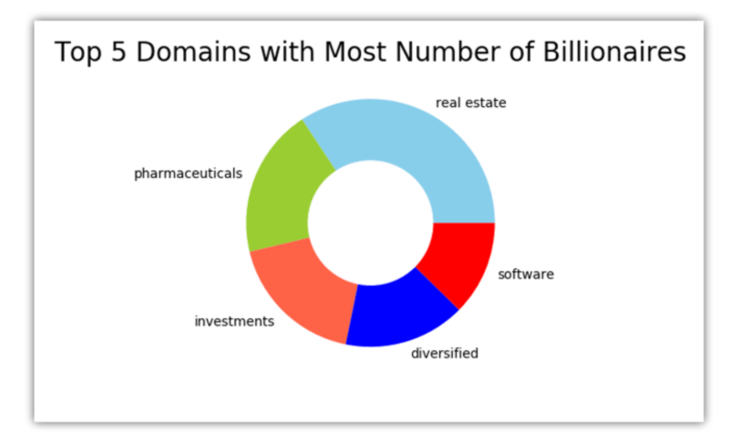
دعنا الآن نلقي نظرة على أفضل 10 مليارديرات وفقًا لصافي الثروة:

```
df = data.sort_values(by = ["NetWorth"],
ascending=False).head(10)
plt.figure(figsize=(20, 10))
sns.histplot(x="Name", hue="NetWorth", data=df)
plt.show()
```



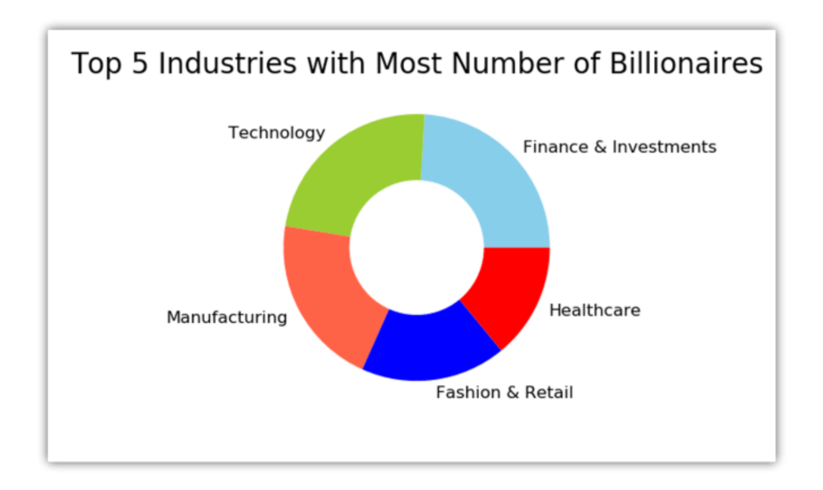
دعنا الآن نلقي نظرة على أهم 5 نطاقات (domains) تضم أكبر عدد من أصحاب المليارات:

```
a = data["Source"].value_counts().head()
index = a.index
sources = a.values
custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(sources, labels=index, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Top 5 Domains to Become a Billionaire",
fontsize=20)
plt.show()
```



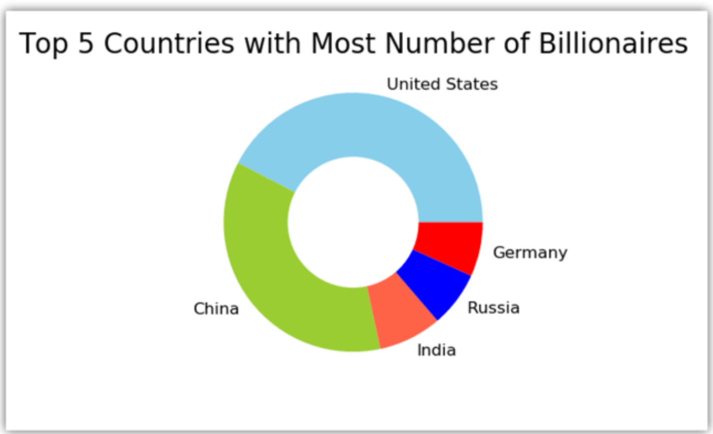
دعنا الآن نلقي نظرة على أفضل 5 صناعات (industries) تضم أكبر عدد من أصحاب المليارات:

```
a = data["Industry"].value_counts().head()
index = a.index
industries = a.values
custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(industries, labels=index, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Top 5 Industries with Most Number of Billionaires",
fontsize=20)
plt.show()
```



دعنا الآن نلقي نظرة على أفضل 5 دول (Countries) بها أكبر عدد من أصحاب المليارات:

```
a = data["Country"].value_counts().head()
index = a.index
Countries = a.values
custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(Countries, labels=index, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Top 5 Countries with Most Number of Billionaires",
fontsize=20)
plt.show()
```



يوضح التصوير أعلاه أن الولايات المتحدة والصين هما البلدان التي يصبح معظم الناس من أصحاب المليارات. وهذا يعني أن بيئة الأعمال ومعدل نجاح بدء التشغيل جيد حقًافي الولايات المتحدة والصين مقارنة ببقية العالم.

الملخص

إذن هذه هي الطريقة التي يمكنك بها العثور على أنماط بين المليارديرات حول العالم لتحليل بيئة الأعمالفي البلدان. يعتمد نجاح أي شركة أو شركة ناشئة كثيرًا على بيئة الأعمالفي بلد ما.في نهاية تحليل المليارديرات العالميين، وجدت أن الصين والولايات المتحدة هما الدولتان اللتان لديهما أكبر عدد من المليارديرات، الأمر الذي يخلص إلى أن بيئة الأعمال ومعدل نجاح شركة ناشئة أفضل بكثيرفي الولايات المتحدة والصين مقارنة بالبقية. من العالم. آمل أن تكون قد أحببت هذا المقال عن تحليل المليارديرات باستخدام بايثون.

28) تحليل البطالة مع بايثون Unemployment Analysis with Python

تُقاس البطالة (Unemployment) من خلال معدل البطالة (unemployment rate) وهو عدد الأشخاص العاطلين عن العمل كنسبة مئوية من إجمالي القوى العاملة. لقد رأينا زيادة حادة في معدل البطالة خلال Covid-19، لذا فإن تحليل معدل البطالة يمكن أن يكون مشروعًا جيدًا لعلم البيانات.في هذا المقالة، سوف أخوضكفي مهمة تحليل البطالة مع بايثون.

تحليل البطالة مع بايثون

يتم احتساب معدل البطالة على أساس منطقة معينة، لذلك لتحليل البطالة سأستخدم مجموعة بيانات البطالةفي الهند. تحتوي مجموعة البيانات التي أستخدمها هنا على بيانات حول معدل البطالةفي الهند خلال Covid-19. فلنبدأ مهمة تحليل البطالة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/unemployment.csv")
print(data.head())
```

```
           Region        Date  Frequency  ...  Region.1  longitude  latitude
0  Andhra Pradesh  31-01-2020          M  ...     South    15.9129     79.74
1  Andhra Pradesh  29-02-2020          M  ...     South    15.9129     79.74
2  Andhra Pradesh  31-03-2020          M  ...     South    15.9129     79.74
3  Andhra Pradesh  30-04-2020          M  ...     South    15.9129     79.74
4  Andhra Pradesh  31-05-2020          M  ...     South    15.9129     79.74

[5 rows x 9 columns]
```

دعونا نرى ما إذا كانت مجموعة البيانات هذه تحتوي على قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

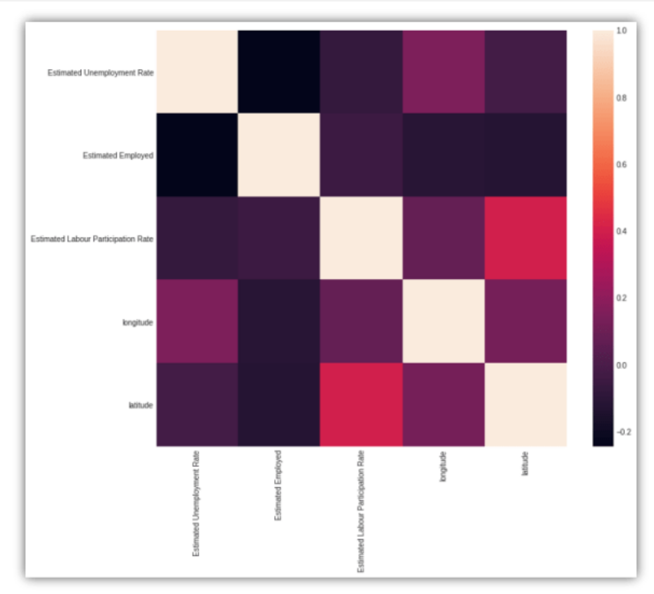
```
Region                                      0
Date                                        0
 Frequency                                  0
 Estimated Unemployment Rate (%)            0
 Estimated Employed                         0
 Estimated Labour Participation Rate (%)    0
Region.1                                    0
longitude                                   0
latitude                                    0
dtype: int64
```

أثناء تحليل القيم المفقودة، وجدت أن أسماء الأعمدة غير صحيحة. لذلك، لفهم هذه البيانات بشكل أفضل، سأعيد تسمية جميع الأعمدة:

```
data.columns= ["States","Date","Frequency",
               "Estimated Unemployment Rate",
               "Estimated Employed",
               "Estimated Labour Participation Rate",
               "Region","longitude","latitude"]
```

دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات هذه:

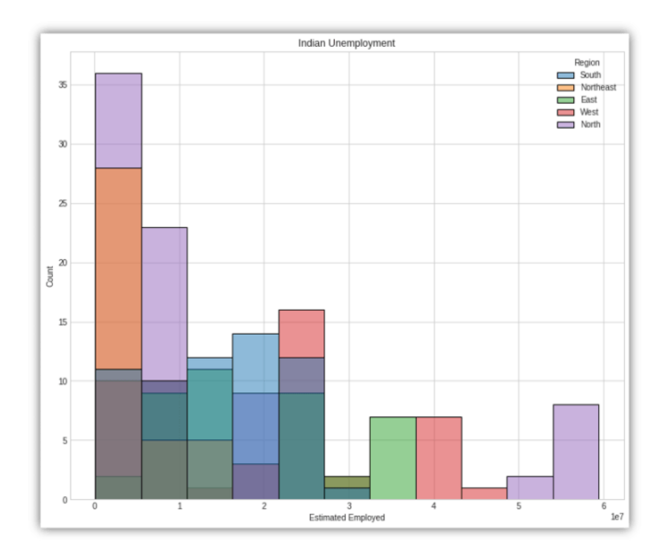
```
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(12, 10))
sns.heatmap(data.corr())
plt.show()
```



تحليل معدل البطالة: تصوير البيانات

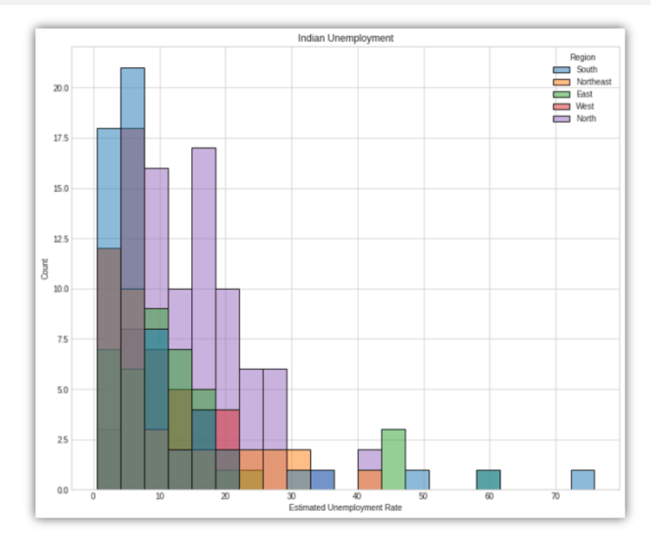
الآن دعنا نتخيل البيانات لتحليل معدل البطالة. سألقي نظرة أولاً على العدد المقدر للموظفين وفقًا لمناطق مختلفة من الهند:

```
data.columns= ["States","Date","Frequency",
               "Estimated Unemployment Rate","Estimated
Employed",
               "Estimated Labour Participation Rate","Region",
               "longitude","latitude"]
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Employed", hue="Region", data=data)
plt.show()
```



دعنا الآن نرى معدل البطالة وفقًا لمناطق مختلفة من الهند:

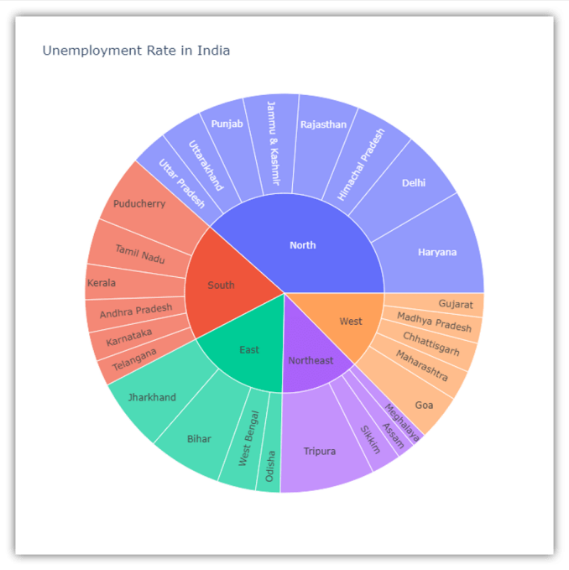
```
plt.figure(figsize=(12, 10))
plt.title("Indian Unemployment")
sns.histplot(x="Estimated Unemployment Rate", hue="Region",
data=data)
plt.show()
```



دعنا الآن ننشئ لوحة تحكم لتحليل معدل البطالة لكل ولاية هندية حسب المنطقة. لهذا، سأستخدم مخطط sunburst :

```
unemploment = data[["States", "Region", "Estimated
Unemployment Rate"]]
figure = px.sunburst(unemploment, path=["Region", "States"],
```

```
                         values="Estimated Unemployment Rate",
                         width=700, height=700,
color_continuous_scale="RdYlGn",
                         title="Unemployment Rate in India")
figure.show()
```



الملخص

هذه هي الطريقة التي يمكنك بها تحليل معدل البطالة باستخدام لغة برمجة بايثون. تُقاس البطالة من خلال معدل البطالة وهو عدد الأشخاص العاطلين عن العمل كنسبة مئوية من إجمالي القوى العاملة. أتمنى أن تكون قد أحببت هذا المقال حول تحليل معدل البطالة باستخدام بايثون.

29) تحليل دردشة WhatsApp مع بايثون WhatsApp Chat Analysis with Python

WhatsApp هو أحد تطبيقات المراسلة الأكثر استخدامًا اليوم مع أكثر من 2 مليار مستخدم حول العالم. تم العثور على أكثر من 65 مليار رسالة يتم إرسالها على WhatsApp يوميًا حتى نتمكن من استخدام محادثات WhatsApp لتحليل الدردشة مع صديق أو عميل أو مجموعة من الأشخاص.في هذه المقالة، سوف أطلعك على مهمة تحليل دردشة WhatsApp مع بايثون.

تحليل دردشة WhatsApp

يمكنك استخدام بيانات WhatsApp الخاصة بك للعديد من مهام علم البيانات مثل تحليل المشاعر (sentiment analysis) واستخراج الكلمات الرئيسية (keyword extraction) والتعرف على الكيانات المسماة (named entity recognition) وتحليل النص والعديد من مهام معالجة اللغة الطبيعية (natural language processing) الأخرى. يعتمد ذلك أيضًا على من تقوم بتحليل رسائل WhatsApp الخاصة بك لأنه يمكنك العثور على الكثير من المعلومات من رسائل WhatsApp الخاصة بك والتي يمكن أن تساعدك أيضًافي حل مشاكل العمل.

قبل البدءفي مهمة تحليل WhatsApp Chat مع بايثون، تحتاج إلى استخراج بيانات WhatsApp الخاصة بك من هاتفك الذكي وهي مهمة سهلة للغاية. لاستخراج محادثات WhatsApp الخاصة بك، فقط افتح أي دردشة مع شخص أو مجموعة واتبع الخطوات المذكورة أدناه:

1. إذا كان لديك جهاز iPhone، فانقر فوق اسم جهة الاتصال (Contact Name) أو اسم المجموعة (Group Name). إذا كان لديك هاتف ذكي يعمل بنظام Android، فانقر فوق النقاط الثلاث أعلاه.
2. ثم قم بالتمرير إلى الأسفل والأعلىفي تصدير الدردشة (Export Chat).
3. ثم حدد بدون وسائط للبساطة إذا سألك ما إذاكنت تريد محادثاتك مع أو بدون وسائط.
4. ثم أرسل هذه الدردشة بالبريد الإلكتروني إلى نفسك وقم بتنزيلها على نظامك.

هذه هي الطريقة التي يمكنك بها بسهولة إجراء محادثات WhatsApp الخاصة بك مع أي شخص أو مجموعة لمهمة تحليل دردشة WhatsApp.في القسم أدناه، سوف آخذك عبر تحليل دردشة WhatsApp مع بايثون.

تحليل دردشة WhatsApp مع بايثون

آمل أن تكون قد فهمت الآن كيفية الحصول على بيانات WhatsApp الخاصة بك لمهمة تحليل دردشة WhatsApp مع بايثون. لنبدأ الآن هذه المهمة عن طريق استيراد مكتبات بايثون الضرورية التي نحتاجها لهذه المهمة:

```
import regex
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator
```

تتطلب مجموعة البيانات التي نستخدمها هنا الكثير من التحضير (preparation)، لذا أقترح عليك إلقاء نظرة على البيانات التي تستخدمها قبل بدء مهمة تحليل الدردشة عبر WhatsApp. نظرًا لأنني قمت بالفعل بتصفح مجموعة البيانات، سأبدأ بكتابة بعض دوال بايثون لإعداد البيانات قبل استيرادها:

```
def date_time(s):
    pattern = '^([0-9]+)(\/)([0-9]+)(\/)([0-9]+), ([0-
9]+):([0-9]+)[ ]?(AM|PM|am|pm)? -'
    result = regex.match(pattern, s)
    if result:
        return True
    return False

def find_author(s):
    s = s.split(":")
    if len(s)==2:
        return True
    else:
        return False

def getDatapoint(line):
    splitline = line.split(' - ')
    dateTime = splitline[0]
    date, time = dateTime.split(", ")
    message = " ".join(splitline[1:])
    if find_author(message):
        splitmessage = message.split(": ")
        author = splitmessage[0]
        message = " ".join(splitmessage[1:])
    else:
        author= None
    return date, time, author, message
```

دعنا الآن نستورد البيانات ونجهزها بطريقة يمكننا استخدامهافي إطار بيانات pandas:

```
data = []
conversation = 'WhatsApp Chat with Sapna.txt'
```

```
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
    messageBuffer = []
    date, time, author = None, None, None
    while True:
        line = fp.readline()
        if not line:
            break
        line = line.strip()
        if date_time(line):
            if len(messageBuffer) > 0:
                data.append([date, time, author, '
'.join(messageBuffer)])
            messageBuffer.clear()
            date, time, author, message = getDatapoint(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)
```

مجموعة البيانات الخاصة بنا جاهزة تمامًا الآن لمهمة تحليل دردشة WhatsApp باستخدام بايثون. دعنا الآن نلقي نظرة على آخر 20 رسالة وبعض الأفكار الأخرى من البيانات:

```
df = pd.DataFrame(data, columns=["Date", 'Time', 'Author',
'Message'])
df['Date'] = pd.to_datetime(df['Date'])
print(df.tail(20))
print(df.info())
print(df.Author.unique())
```

	Date	Time	Author	Message
1285	2021-07-01	8:23 pm	Aman Kharwal	😊
1286	2021-08-01	1:12 am	Sapna	Are you taking the class tomorrow
1287	2021-08-01	1:14 am	Aman Kharwal	No
1288	2021-08-01	1:14 am	Aman Kharwal	Are you?
1289	2021-08-01	10:24 am	Sapna	Hahhahah. 😂😂
1290	2021-08-01	10:31 am	Aman Kharwal	😋
1291	2021-08-01	8:28 pm	None	Aman Kharwal: https://www.youtube.com/watch?v=...
1292	2021-08-01	8:29 pm	Aman Kharwal	hope you like it
1293	2021-08-01	8:43 pm	Sapna	Congratulations 😀😀
1294	2021-08-01	8:43 pm	Sapna	I'll watch it😀
1295	2021-08-01	8:45 pm	Aman Kharwal	Thanks 😊🌹
1296	2021-08-01	8:46 pm	Sapna	😀😀
1297	2021-09-01	10:26 am	Sapna	U.S politics has taken an interesting turn
1298	2021-09-01	10:26 am	Sapna	Fun to watch the mockery
1299	2021-09-01	10:35 am	Aman Kharwal	Yes😂
1300	2021-09-01	4:54 pm	Aman Kharwal	Do you know we can also analyse the whatsapp c...
1301	2021-09-01	4:54 pm	Aman Kharwal	😁
1302	2021-09-01	5:54 pm	Sapna	This is interesting
1303	2021-09-01	5:54 pm	Sapna	Do share the code
1304	2021-09-01	5:54 pm	Sapna	😀😀

الآن دعنا نلقي نظرة على العدد الإجمالي للرسائل بين دردشة WhatsApp هذه:

```
total_messages = df.shape[0]
print(total_messages)
```

```
1288
```

الآن دعنا نستخرج الرموز التعبيرية (emojis) الموجودة بين الدردشات ونلقي نظرة على الرموز التعبيرية الموجودةفي هذه الدردشة:

```
def split_count(text):
    emoji_list = []
    data = regex.findall(r'\X',text)
    for word in data:
        if any(char in emoji.UNICODE_EMOJI for char in word):
            emoji_list.append(word)
    return emoji_list
df['emoji'] = df["Message"].apply(split_count)

emojis = sum(df['emoji'].str.len())
print(emojis)
```

```
367
```

دعنا الآن نستخرج عناوين URL الموجودةفي هذه الدردشة ونلقي نظرة على الإحصاءات النهائية:

```
URLPATTERN = r'(https?://\S+)'
df['urlcount'] = df.Message.apply(lambda x:
regex.findall(URLPATTERN, x)).str.len()
links = np.sum(df.urlcount)

print("Chats between Aman and Sapna")
print("Total Messages: ", total_messages)
print("Number of Media Shared: ", media_messages)
print("Number of Emojis Shared", emojis)
print("Number of Links Shared", links)
```

```
Chats between Aman and Sapna
Total Messages:  1288
Number of Media Shared:  11
Number of Emojis Shared 367
Number of Links Shared 1
```

دعنا الآن نجهز هذه البيانات للحصول على مزيد من الأفكار لتحليل جميع الرسائل المرسلةفي هذه الدردشة بمزيد من التفصيل:

```
media_messages_df = df[df['Message'] == '<Media omitted>']
messages_df = df.drop(media_messages_df.index)
messages_df['Letter_Count'] =
messages_df['Message'].apply(lambda s : len(s))
```

```
messages_df['Word_Count'] =
messages_df['Message'].apply(lambda s : len(s.split(' ')))
messages_df["MessageCount"]=1

l = ["Aman Kharwal", "Sapna"]
for i in range(len(l)):
  # Filtering out messages of particular user
  req_df= messages_df[messages_df["Author"] == l[i]]
  # req_df will contain messages of only one particular user
  print(f'Stats of {l[i]} -')
  # shape will print number of rows which indirectly means the
number of messages
  print('Messages Sent', req_df.shape[0])
  #Word_Count contains of total words in one message. Sum of
all words/ Total Messages will yield words per message
  words_per_message =
(np.sum(req_df['Word_Count']))/req_df.shape[0]
  print('Average Words per message', words_per_message)
  #media conists of media messages
  media = media_messages_df[media_messages_df['Author'] ==
l[i]].shape[0]
  print('Media Messages Sent', media)
  # emojis conists of total emojis
  emojis = sum(req_df['emoji'].str.len())
  print('Emojis Sent', emojis)
  #links consist of total links
  links = sum(req_df["urlcount"])
  print('Links Sent', links)
```

```
Stats of Aman Kharwal -
Messages Sent 687
Average Words per message 6.165938864628821
Media Messages Sent 9
Emojis Sent 228
Links Sent 1

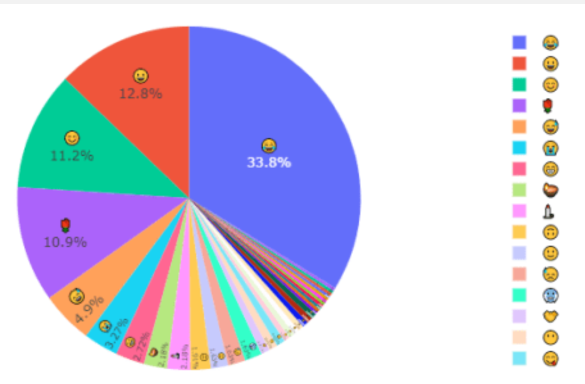
Stats of Sapna -
Messages Sent 590
Average Words per message 6.3830508474576275
Media Messages Sent 2
Emojis Sent 139
Links Sent 0
```

```
total_emojis_list = list(set([a for b in messages_df.emoji for
a in b]))
total_emojis = len(total_emojis_list)

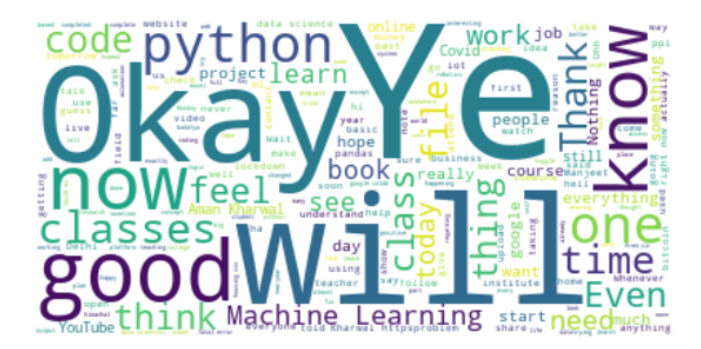
total_emojis_list = list([a for b in messages_df.emoji for a
in b])
emoji_dict = dict(Counter(total_emojis_list))
emoji_dict = sorted(emoji_dict.items(), key=lambda x: x[1],
reverse=True)
for i in emoji_dict:
  print(i)

emoji_df = pd.DataFrame(emoji_dict, columns=['emoji',
'count'])
```

```
import plotly.express as px
fig = px.pie(emoji_df, values='count', names='emoji')
fig.update_traces(textposition='inside',
textinfo='percent+label')
fig.show()
```

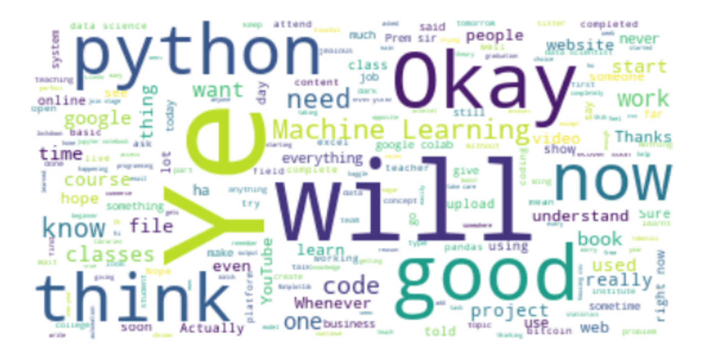


دعنا الآن نلقي نظرة على الكلمات الأكثر استخدامًافي دردشة WhatsApp هذه من خلال رسم سحابة الكلمات (word cloud):

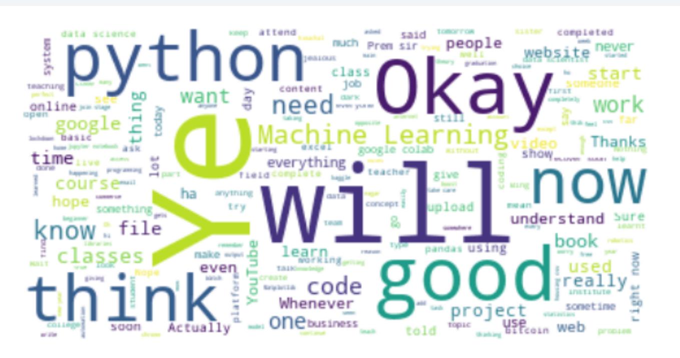


دعنا الآن نلقي نظرة على الكلمات الأكثر استخدامًا من قبل كل شخص من خلال تخيل سحابتين مختلفتين من الكلمات:

```
l = ["Aman Kharwal", "Sapna"]
for i in range(len(l)):
  dummy_df = messages_df[messages_df['Author'] == l[i]]
  text = " ".join(review for review in dummy_df.Message)
  stopwords = set(STOPWORDS)
  #Generate a word cloud image
  print('Author name',l[i])
  wordcloud = WordCloud(stopwords=stopwords,
background_color="white").generate(text)
  #Display the generated image
  plt.figure( figsize=(10,5))
  plt.imshow(wordcloud, interpolation='bilinear')
  plt.axis("off")
  plt.show()
```




```
Author name Aman Kharwal
```




```
Author name Sapna
```

الملخص

هذه هي الطريقة التي يمكننا بها بسهولة تحليل أي دردشة WhatsApp بينك وبين صديقك أو عميلك أو حتى مجموعة من الأشخاص. يمكنك أيضًا استخدام هذه البياناتفي العديد من المهام الأخرى الخاصة بمعالجة اللغة الطبيعية. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة تحليل دردشة WhatsApp مع بايثون.

30) تحليل المشاعر في دردشة WhatsApp باستخدام بايثون WhatsApp Chat Sentiment Analysis using Python

يعد WhatsApp مصدرًا رائعًا للبيانات لتحليل العديد من الأنماط والعلاقات بين شخصين أو أكثر يتحدثون شخصيًا أو حتى في مجموعات. إذا كنت تريد أن تعرف كيف يمكننا تحليل مشاعر دردشة WhatsApp، فهذه المقالة مناسبة لك.في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعرفي دردشة WhatsApp باستخدام بايثون.

تحليل المشاعر دردشة WhatsApp

لتحليل مشاعر دردشة WhatsApp، نحتاج إلى جمع البيانات من WhatsApp. يجب أن يستخدم معظمكم تطبيق المراسلة هذا، لذلك لجمع البيانات حول الدردشة، ما عليك سوى اتباع الخطوات المذكورة أدناه:

1. للآيفون:
    1. افتح الدردشة الخاصة بك مع شخص أو مجموعة.
    2. فقط اضغط على ملف تعريف الشخص أو المجموعة.
    3. سترى خيارًا لتصدير الدردشة بالأسفل.
4. لأجهزة الأندرويد:
    1. افتح الدردشة الخاصة بك مع شخص أو مجموعة.
    2. انقر فوق النقاط الثلاث أعلاه.
    3. انقر فوق المزيد.
    4. انقر فوق دردشة التصدير.

سترى خيارًا لإرفاق الوسائط أثناء تصدير الدردشة. من أجل البساطة، من الأفضل عدم إرفاق الوسائط. أخيرًا، أدخل بريدك الإلكتروني وستجد دردشة WhatsApp في صندوق الوارد الخاص بك.

تحليل مشاعر الدردشة عبر WhatsApp باستخدام بايثون

لنبدأ الآن بمهمة تحليل مشاعر الدردشة عبر WhatsApp باستخدام بايثون. سأبدأ هذه المهمة بتحديد بعض الوظائف المساعدة لأن البيانات التي نحصل عليها من WhatsApp ليست مجموعة بيانات جاهزة للاستخدام في أي نوع من مهام علم البيانات. لذلك، لإعداد بياناتك لمهمة تحليل المشاعر، ما عليك سوى تحديد جميع الدوال على النحو المحدد أدناه:

```
import re
import pandas as pd
import numpy as np
import emoji
from collections import Counter
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS,
ImageColorGenerator

# Extract Time
def date_time(s):
    pattern = '^([0-9]+)(\/)([0-9]+)(\/)([0-9]+), ([0-
9]+):([0-9]+)[ ]?(AM|PM|am|pm)? -'
    result = re.match(pattern, s)
    if result:
        return True
    return False

# Find Authors or Contacts
def find_author(s):
    s = s.split(":")
    if len(s)==2:
        return True
    else:
        return False

# Finding Messages
def getDatapoint(line):
    splitline = line.split(' - ')
    dateTime = splitline[0]
    date, time = dateTime.split(", ")
    message = " ".join(splitline[1:])
    if find_author(message):
        splitmessage = message.split(": ")
        author = splitmessage[0]
        message = " ".join(splitmessage[1:])
    else:
        author= None
    return date, time, author, message
```

لا يهم إذا كنت تستخدم مجموعة بيانات دردشة جماعية أو محادثتك مع شخص واحد. ستعمل جميع الدوال المحددة أعلاه على إعداد بياناتك لمهمة تحليل المشاعر وأي مهمة تتعلق بعلم

البيانات. الآن إليك كيف يمكننا إعداد البيانات التي جمعناها من WhatsApp باستخدام الدوال المذكورة أعلاه:

```
data = []
conversation = 'WhatsApp Chat with Sapna.txt'
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
    messageBuffer = []
    date, time, author = None, None, None
    while True:
        line = fp.readline()
        if not line:
            break
        line = line.strip()
        if date_time(line):
            if len(messageBuffer) > 0:
                data.append([date, time, author, '
'.join(messageBuffer)])
            messageBuffer.clear()
            date, time, author, message = getDatapoint(line)
            messageBuffer.append(message)
        else:
            messageBuffer.append(line)
```

الآن إليك كيف يمكننا تحليل مشاعر دردشة WhatsApp باستخدام بايثون:

```
df = pd.DataFrame(data, columns=["Date", 'Time', 'Author',
'Message'])
df['Date'] = pd.to_datetime(df['Date'])

data = df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Message"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Message"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Message"]]
print(data.head())
```

	Date	Time	Author	...	Positive	Negative	Neutral
0	2020-04-06	12:30 pm	Sapna	...	0.0	0.000	1.000
1	2020-04-06	12:30 pm	Sapna	...	0.0	0.000	1.000
2	2020-04-06	12:54 pm	Aman Kharwal	...	0.0	0.000	1.000
3	2020-04-06	12:55 pm	Sapna	...	0.0	0.383	0.617
4	2020-04-06	1:00 pm	Aman Kharwal	...	0.0	0.000	1.000

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
```

```
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Output:
Neutral 🙂
```

لذا، فإن البيانات التي استخدمتها تشير إلى أن معظم الرسائل بيني وبين الشخص الآخر محايدة (neutral). مما يعني أنه ليس إيجابيًا (positive) أو سلبيًا (negative).

الملخص

إذن هذه هي الطريقة التي يمكننا بها أداء مهمة تحليل المشاعرفي دردشة WhatsApp. يعد WhatsApp مصدرًا رائعًا للبيانات لمهمة تحليل المشاعر وكل مهمةفي علم البيانات تعتمد على معالجة اللغة الطبيعية. آمل أن تكون قد أحببت هذه المقالة حول مهمة تحليل المشاعرفي دردشة WhatsApp باستخدام بايثون.

32) تحليل لقاحات Covid-19 باستخدام بايثون Covid-19 Vaccines Analysis with Python

كان هناك وقت خرج فيه Covid-19 عن السيطرة. حتى بعد الإغلاق، أدى ذلك إلى زيادة سريعةفي الحالات حيث تمت السيطرة على بعض الحالاتفي بعض البلدان ولكن تمت التضحية بالاقتصاد.في مثل هذه الحالة، يُنظر إلى اللقاحات (vaccines) فقط على أنها الأداة الوحيدة التي يمكن أن تساعد العالمفي مكافحة فيروس كورونا.في هذه المقالة، سوف أطلعك على مهمة تحليل لقاحات Covid-19 باستخدام بايثون.

تحليل لقاحات Covid-19

تم تقديم العديد من اللقاحات حتى الآن لمحاربة Covid-19. لا يوجد لقاح يضمن دقة 100٪ حتى الآن، لكن معظم الشركات المصنعة تدعي أن لقاحها ليس دقيقًا بنسبة 100٪، ولكن مع ذلك، فإنه سينقذ حياتك من خلال إعطائك مناعة.

وهكذا، يحاول كل بلد تطعيم جزء كبير من سكانه حتى لا يعتمد على لقاح واحد. هذا ما سأقوم بتحليلهفي هذه المقالة، وهو عدد اللقاحات التي تستخدمها كل دولة لمحاربة Covid-19.في القسم أدناه، سوف آخذك من خلال برنامج تعليمي لعلم البيانات حول تحليل لقاحات Covid-19 باستخدام بايثون.

تحليل لقاحات Covid-19 باستخدام بايثون

مجموعة البيانات التي سأستخدمها هنا لمهمة تحليل لقاحات covid-19 مأخوذة من Kaggle. لنبدأ باستيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("country_vaccinations.csv")
data.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0

دعنا الآن نستكشف هذه البيانات قبل أن نبدأفي تحليل اللقاحات التي تأخذها البلدان:

```
data.describe()
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred
count	6.575000e+03	5.926000e+03	4.223000e+03	5.507000e+03	1.069500e+04	6575.000000
mean	3.630372e+06	2.585810e+06	1.175809e+06	1.192650e+05	7.048124e+04	12.295927
std	1.396665e+07	9.013711e+06	4.860110e+06	4.420816e+05	2.988761e+05	20.384869
min	0.000000e+00	0.000000e+00	1.000000e+00	-2.928600e+04	0.000000e+00	0.000000
25%	5.159850e+04	4.880650e+04	2.208100e+04	3.171500e+03	9.875000e+02	0.970000
50%	3.459940e+05	2.925895e+05	1.474380e+05	1.530100e+04	6.001000e+03	4.720000
75%	1.605542e+06	1.199198e+06	5.776635e+05	5.960150e+04	2.799800e+04	14.615000
max	1.834677e+08	1.171429e+08	7.069264e+07	7.185000e+06	5.190143e+06	188.990000

```
pd.to_datetime(data.date)
data.country.value_counts()
```

```
United Kingdom      118
Northern Ireland    118
Wales               118
England             118
Canada              118
                   ...
Mali                  4
Bahamas               2
Brunei                2
Laos                  1
Armenia               1
Name: country, Length: 175, dtype: int64
```

تتكون المملكة المتحدة من إنجلترا واسكتلندا وويلز وأيرلندا الشمالية. ولكن في البيانات أعلاه، تم ذكر هذه البلدان بشكل منفصل بنفس القيم كما في المملكة المتحدة. لذلك قد يكون هذا خطأ أثناء تسجيل هذه البيانات. لذلك دعونا نرى كيف يمكننا إصلاح هذا الخطأ:

```
data = data[data.country.apply(lambda x: x not in ["England",
"Scotland", "Wales", "Northern Ireland"])]
data.country.value_counts()
```

```
Canada            118
United Kingdom    118
China             117
Russia            117
Israel            113
                 ...
Mali                4
Bahamas             2
Brunei              2
Laos                1
Armenia             1
Name: country, Length: 171, dtype: int64
```

دعنا الآن نستكشف اللقاحات المتوفرة في مجموعة البيانات هذه:

```
data.vaccines.value_counts()
```

```
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech                                           2587
Oxford/AstraZeneca                                                                     1673
Pfizer/BioNTech                                                                        1416
Oxford/AstraZeneca, Pfizer/BioNTech                                                     845
Pfizer/BioNTech, Sinovac                                                                475
Moderna, Pfizer/BioNTech                                                                407
Sputnik V                                                                               351
Oxford/AstraZeneca, Sinovac                                                             301
Oxford/AstraZeneca, Sinopharm/Beijing                                                   268
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V                                        235
Pfizer/BioNTech, Sinopharm/Beijing                                                      208
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V                       202
Sinopharm/Beijing                                                                       186
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac                                            123
Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac                                             117
EpiVacCorona, Sputnik V                                                                 117
Johnson&Johnson, Moderna, Pfizer/BioNTech                                               112
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V                                 108
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V              104
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V       96
Covaxin, Oxford/AstraZeneca                                                              86
Sinovac                                                                                  84
Moderna, Oxford/AstraZeneca                                                              79
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac                                           61
Johnson&Johnson                                                                          54
Sinopharm/Beijing, Sputnik V                                                             51
Pfizer/BioNTech, Sputnik V                                                               42
Pfizer/BioNTech, Sinovac, Sputnik V                                                      29
Name: vaccines, dtype: int64
```

إذن لدينا جميع لقاحات Covid-19 تقريبًا متوفرة في مجموعة البيانات هذه. الآن سوف أقوم بإنشاء إطار بيانات جديد من خلال اختيار أعمدة اللقاح (vaccine) والدولة (country) فقط لاستكشاف اللقاح الذي يتم تناوله في أي دولة:

```
df = data[["vaccines", "country"]]
df.head()
```

	vaccines	country
0	Oxford/AstraZeneca	Afghanistan
1	Oxford/AstraZeneca	Afghanistan
2	Oxford/AstraZeneca	Afghanistan
3	Oxford/AstraZeneca	Afghanistan
4	Oxford/AstraZeneca	Afghanistan

دعنا الآن نرى عدد البلدان التي تأخذ كل اللقاحات المذكورة في هذه البيانات:

```
dict_ = {}
for i in df.vaccines.unique():
  dict_[i] = [df["country"][j] for j in
df[df["vaccines"]==i].index]
```

```
vaccines = {}
for key, value in dict_.items():
  vaccines[key] = set(value)
for i, j in vaccines.items():
  print(f"{i}:>>{j}")
```

```
Oxford/AstraZeneca:>>{'Gambia', 'Maldives', 'Jamaica', 'Saint Lucia', 'Myanmar', 'Barbados',
'Brunei', 'Togo', 'Ghana', 'Mauritius', 'Malawi', 'Antigua and Barbuda', 'Nepal', 'Taiwan', 'Uganda',
'Bahamas', 'Vietnam', 'Eswatini', 'Saint Helena', 'Mongolia', 'Kenya', "Cote d'Ivoire", 'Moldova',
'Trinidad and Tobago', 'Uzbekistan', 'Mali', 'Botswana', 'Bangladesh', 'Falkland Islands',
'Suriname', 'Ukraine', 'Papua New Guinea', 'Afghanistan', 'Sierra Leone', 'Sudan', 'Nigeria',
'Belize', 'Grenada', 'Montserrat', 'Kosovo', 'Sri Lanka', 'Georgia', 'Bhutan', 'Saint Kitts and
Nevis', 'Solomon Islands', 'Angola', 'Guyana', 'Sao Tome and Principe', 'Cape Verde', 'Dominica',
'Saint Vincent and the Grenadines', 'Anguilla'}
Pfizer/BioNTech, Sinovac:>>{'Colombia', 'Malaysia', 'Uruguay', 'Albania', 'Chile', 'Turkey', 'Hong
Kong'}
Sputnik V:>>{'Guinea', 'Iran', 'Paraguay', 'Kazakhstan', 'Algeria', 'Syria', 'Belarus', 'Armenia',
'Venezuela'}
Pfizer/BioNTech:>>{'Cayman Islands', 'Greenland', 'Monaco', 'Andorra', 'Gibraltar', 'Lebanon', 'Turks
and Caicos Islands', 'Panama', 'Japan', 'North Macedonia', 'Qatar', 'Cyprus', 'Bermuda', 'Slovakia',
'Ecuador', 'New Zealand', 'Costa Rica', 'Kuwait'}
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V:>>{'Argentina', 'Pakistan', 'Bolivia'}
Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Australia', 'United Kingdom', 'Jersey', 'Isle of Man', 'South
Korea', 'Sweden', 'Slovenia', 'Guernsey', 'Oman', 'Saudi Arabia'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>>{'Austria', 'Rwanda', 'Canada', 'Belgium',
'Lithuania', 'France', 'Germany', 'Spain', 'Latvia', 'Estonia', 'Ireland', 'Norway', 'Luxembourg',
'Netherlands', 'Italy', 'Malta', 'Czechia', 'Iceland', 'Palestine', 'Croatia', 'Denmark', 'Greece',
'Poland', 'Romania', 'Bulgaria', 'Portugal', 'Finland'}
Sinovac:>>{'Azerbaijan'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>>{'Serbia', 'Bahrain'}
Oxford/AstraZeneca, Sinovac:>>{'Brazil', 'Dominican Republic', 'Indonesia', 'Philippines',
'Thailand'}
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac:>>{'Cambodia'}
Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac:>>{'China'}
Oxford/AstraZeneca, Sinopharm/Beijing:>>{'Iraq', 'Morocco', 'Seychelles', 'Egypt'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>>{'Northern Cyprus', 'El Salvador'}
Sinopharm/Beijing:>>{'Equatorial Guinea', 'Mauritania', 'Senegal', 'Zimbabwe', 'Gabon', 'Kyrgyzstan',
'Namibia', 'Mozambique'}
Moderna, Pfizer/BioNTech:>>{'Liechtenstein', 'Faeroe Islands', 'Israel', 'Switzerland', 'Singapore'}
Moderna, Oxford/AstraZeneca:>>{'Honduras', 'Guatemala'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>>{'Hungary'}
Covaxin, Oxford/AstraZeneca:>>{'India'}
Pfizer/BioNTech, Sinopharm/Beijing:>>{'Peru', 'Macao', 'Jordan'}
Sinopharm/Beijing, Sputnik V:>>{'Montenegro', 'Laos'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>>{'Mexico'}
EpiVacCorona, Sputnik V:>>{'Russia'}
Pfizer/BioNTech, Sputnik V:>>{'San Marino'}
Johnson&Johnson:>>{'South Africa'}
Pfizer/BioNTech, Sinovac, Sputnik V:>>{'Tunisia'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V:>>{'United Arab
Emirates'}
Johnson&Johnson, Moderna, Pfizer/BioNTech:>>{'United States'}
```

الآن دعنا نتخيل هذه البيانات لإلقاء نظرة على مجموعة اللقاحات التي يستخدمها كل بلد:

```
import plotly.express as px
import plotly.offline as py
```

```
vaccine_map = px.choropleth(data, locations = 'iso_code',
color = 'vaccines')
vaccine_map.update_layout(height=300,
margin={"r":0,"t":0,"l":0,"b":0})
vaccine_map.show()
```

vaccines
Oxford/AstraZeneca
Pfizer/BioNTech, Sinovac
Sputnik V
Pfizer/BioNTech
Oxford/AstraZeneca, Sinopharm/Beijir
Oxford/AstraZeneca, Pfizer/BioNTech
Moderna, Oxford/AstraZeneca, Pfizer/
Sinovac
Oxford/AstraZeneca, Pfizer/BioNTech,
Oxford/AstraZeneca, Sinovac
Oxford/AstraZeneca, Sinopharm/Beijir
Sinopharm/Beijing, Sinopharm/Wuhar
Oxford/AstraZeneca, Sinopharm/Beijir
Oxford/AstraZeneca, Pfizer/BioNTech,
Sinopharm/Beijing

الملخص

إذن هذه هي الطريقة التي يمكننا بها تحليل نوع اللقاحات التي يأخذها كل بلد اليوم. يمكنك استكشاف المزيد من الأفكار من مجموعة البيانات هذه حيث يوجد الكثير الذي يمكنك القيام به باستخدام هذه البيانات. آمل أن تكون قد أحببت هذا المقال حول تحليل لقاحات Covid-19 باستخدام بايثون.

33) التنبؤ بمبيعات ألعاب الفيديو باستخدام لغة بايثون
Video Game Sales Prediction with Python

تحليل مبيعات ألعاب الفيديو (Video game sales analysis) هو بيان مشكلة شائع على Kaggle. يمكنك العمل على هذه المشكلة لتحليل مبيعات أكثر من 16500 لعبة أو يمكنك أيضًا تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو. لذلك إذا كنت تريد معرفة كيفية تدريب نموذج تنبؤ بمبيعات ألعاب الفيديو، فهذه المقالة مناسبة لك.في هذه المقالة، سأوجهك خلال مهمة تعلم الآلة حول تدريب نموذج توقع مبيعات لعبة فيديو باستخدام بايثون.

نموذج التنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون

تحليل بيانات المبيعات لأكثر من 16500 لعبة هو بيان مشكلة شائع جدًا على Kaggle. يمكنك إما حل هذه المشكلة للعثور على العديد من الأنماط (patterns) والعلاقات (relationships) بين العوامل التي تؤثر على مبيعات ألعاب الفيديو، أو يمكنك استخدام مجموعة البيانات هذه للتنبؤ بمبيعات ألعاب الفيديوفي المستقبل. لذلكفي القسم أدناه، سأوجهك خلال كيفية تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون.

تحتوي مجموعة البيانات التي أستخدمها لهذه المهمة على قائمة بألعاب الفيديو ومبيعاتها. لنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("vgsales.csv")
print(data.head())
```

```
   Rank                      Name Platform    Year  ... EU_Sales JP_Sales  Other_Sales  Global_Sales
0     1                Wii Sports      Wii  2006.0  ...    29.02     3.77         8.46         82.74
1     2         Super Mario Bros.      NES  1985.0  ...     3.58     6.81         0.77         40.24
2     3            Mario Kart Wii      Wii  2008.0  ...    12.88     3.79         3.31         35.82
3     4         Wii Sports Resort      Wii  2009.0  ...    11.01     3.28         2.96         33.00
4     5  Pokemon Red/Pokemon Blue       GB  1996.0  ...     8.89    10.22         1.00         31.37
```

دعنا الآن نرى ما إذا كانت مجموعة البيانات هذه تحتوي على قيم خالية:

```
print(data.isnull().sum())
```

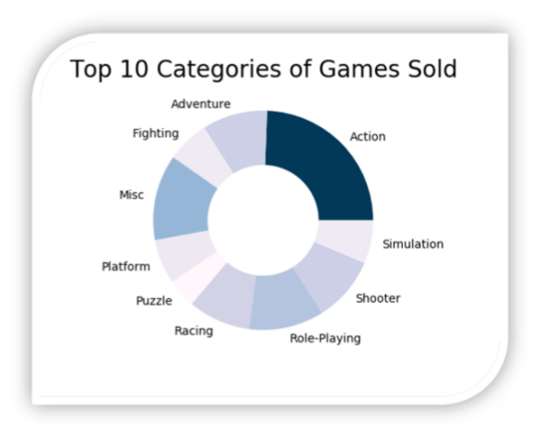
```
Rank              0
Name              0
Platform          0
Year            271
Genre             0
Publisher        58
NA_Sales          0
EU_Sales          0
JP_Sales          0
Other_Sales       0
Global_Sales      0
dtype: int64
```

سأقوم الآن بإنشاء مجموعة بيانات جديدة تزيل القيم الخالية:

```
data = data.dropna()
```

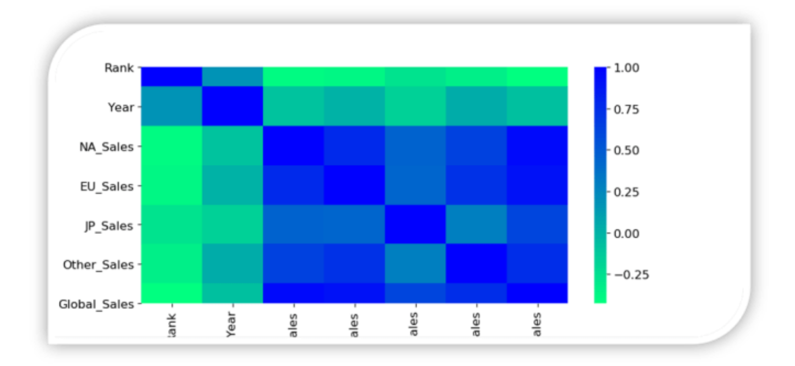
قبل تدريب النموذج، دعنا نلقي نظرة على أفضل 10 فئات من الألعاب مبيعًا:

```
import matplotlib as mpl
game = data.groupby("Genre")["Global_Sales"].count().head(10)
custom_colors = mpl.colors.Normalize(vmin=min(game),
vmax=max(game))
colours = [mpl.cm.PuBu(custom_colors(i)) for i in game]
plt.figure(figsize=(7,7))
plt.pie(game, labels=game.index, colors=colours)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Top 10 Categories of Games Sold", fontsize=20)
plt.show()
```



دعنا الآن نلقي نظرة على الارتباط (correlation) بين ميزات مجموعة البيانات هذه:

```
print(data.corr())
sns.heatmap(data.corr(), cmap="winter_r")
plt.show()
```



نموذج التنبؤ بمبيعات ألعاب الفيديو التدريبية

دعنا الآن نرى كيفية تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون . سأقوم بإعداد البيانات عن طريق تخزين الميزات التي نحتاجها لتدريب هذا النموذجفي المتغير xوتخزين العمود الهدففي المتغيرy :

```
x = data[["Rank", "NA_Sales", "EU_Sales", "JP_Sales",
"Other_Sales"]]
y = data["Global_Sales"]
```

الآن دعنا نقسم البيانات ونستخدم خوارزمية الانحدار الخطي (linear regression) لتدريب هذا النموذج:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
test_size=0.2, random_state=42)

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(xtrain, ytrain)
predictions = model.predict(xtest)
```

الملخص

هذه هي الطريقة التي يمكننا بها تدريب نموذج التعلم الآلي للتنبؤ بمبيعات ألعاب الفيديو. هذا بيان مشهور لمشكلة Kaggle يمكنك استخدامه لتحسين مهاراتكفي العمل مع البيانات والتدريب على نموذج التعلم الآلي. آمل أن تكون قد أحببت هذه المقالة حول كيفية تدريب نموذج التنبؤ بمبيعات ألعاب الفيديو باستخدام بايثون.

34) تحليل رحلات Uber باستخدام بايثون Uber Trips Analysis using Python

كانت Uber مصدرًا رئيسيًا للسفر للأشخاص الذين يعيشون في المناطق الحضرية. بعض الناس لا يملكون سياراتهم بينما البعض الآخر لا يقود سياراتهم عمداً بسبب جدولهم المزدحم. لذلك تستخدم أنواع مختلفة من الناس خدمات Uber وخدمات سيارات الأجرة الأخرى.في هذه المقالة، سوف آخذك إلى تحليل رحلات Uber باستخدام بايثون.

تحليل رحلات Uber

من خلال تحليل رحلات Uber، يمكننا رسم العديد من الأنماط، مثل أي يوم يحتوي على أعلى وأقل عدد أو أكثر الساعات ازدحامًا لـ Uber والعديد من الأنماط الأخرى. تستند مجموعة البيانات التي أستخدمها هنا إلى رحلات Uber من نيويورك، وهي مدينة بها نظام نقل معقد للغاية مع مجتمع سكني كبير.

تحتوي مجموعة البيانات على بيانات لحوالي 4.5 مليون بيك أب في مدينة نيويورك من أبريل إلى سبتمبر و 14.3 مليون بيك أب من يناير إلى يونيو 2015. يمكنك فعل الكثير باستخدام مجموعة البيانات هذه بدلاً من مجرد تحليلها. لكن في الوقت الحالي، في القسم أدناه، سوف آخذك إلى تحليل رحلات Uber باستخدام بايثون.

تحليل رحلات Uber باستخدام بايثون

سأبدأ مهمة تحليل رحلات Uber عن طريق استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = pd.read_csv("uber.csv")
data["Date/Time"] = data["Date/Time"].map(pd.to_datetime)
data.head()
```

	Date/Time	Lat	Lon	Base
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512

تحتوي هذه البيانات على بيانات حول التاريخ والوقت وخط العرض وخط الطول وعمود أساسي يحتوي على رمز تابع لـ uber pickup. يمكنك الحصول على المزيد من مجموعات البيانات

لمهمة تحليل رحلات Uber من هنا، في الوقت الحالي، دعنا نجهز البيانات التي أستخدمها هنا لتحليل رحلات Uber وفقًا للأيام والساعات:

```
data["Day"] = data["Date/Time"].apply(lambda x: x.day)
data["Weekday"] = data["Date/Time"].apply(lambda x:
x.weekday())
data["Hour"] = data["Date/Time"].apply(lambda x: x.hour)
print(data.head())
```

	Date/Time	Lat	Lon	Base	Day	Weekday	Hour
0	2014-09-01 00:01:00	40.2201	-74.0021	B02512	1	0	0
1	2014-09-01 00:01:00	40.7500	-74.0027	B02512	1	0	0
2	2014-09-01 00:03:00	40.7559	-73.9864	B02512	1	0	0
3	2014-09-01 00:06:00	40.7450	-73.9889	B02512	1	0	0
4	2014-09-01 00:11:00	40.8145	-73.9444	B02512	1	0	0

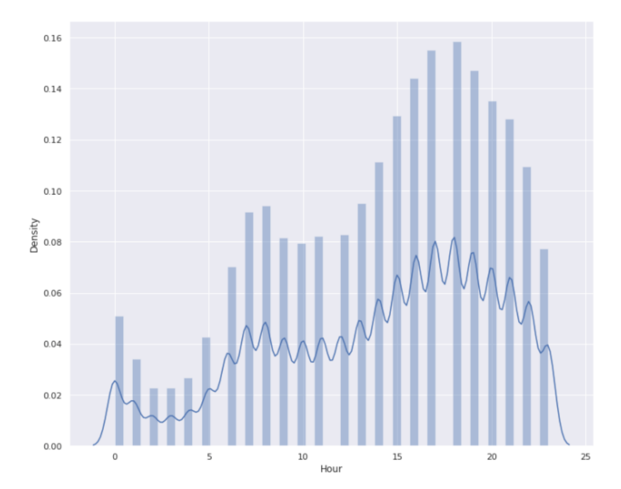
لذلك أعددت هذه البيانات وفقًا للأيام والساعات، حيث إنني أستخدم رحلات Uber لشهر سبتمبر، لذلك دعونا نلقي نظرة على كل يوم لمعرفة اليوم الذي كانت فيه رحلات Uber هي الأعلى:

```
sns.set(rc={'figure.figsize(10 ,12):'})
sns.distplot(data["Day"])
```



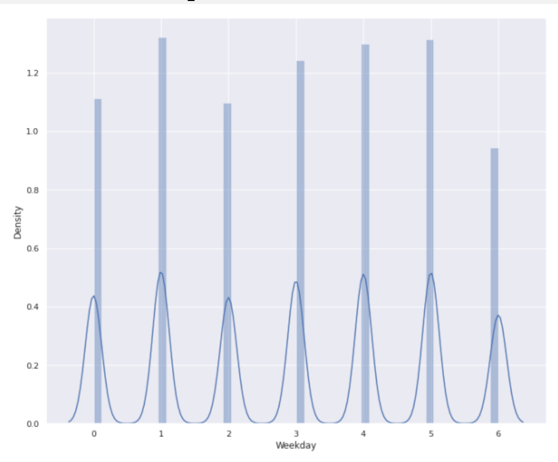
من خلال النظر إلى الرحلات اليومية، يمكننا القول إن رحلات Uber ترتفع في أيام العمل وتنخفض في عطلات نهاية الأسبوع. الآن دعنا نحلل رحلات Uber وفقًا للساعات:

```
sns.distplot(data["Hour"])
```



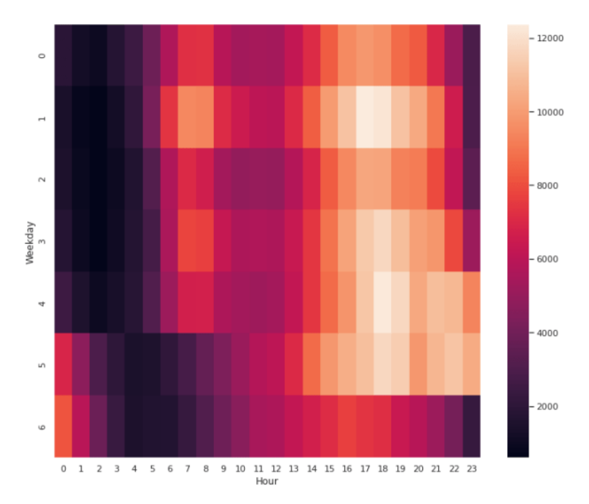
وفقًا لبيانات كل ساعة، تقل رحلات Uber بعد منتصف الليل ثم تبدأفي الزيادة بعد الساعة 5 صباحًا وتستمر الرحلاتفي الارتفاع حتى الساعة 6 مساءً بحيث تكون الساعة 6 مساءً هي أكثر الساعات ازدحامًا لـ Uber ثم تبدأ الرحلاتفي التناقص. الآن دعنا نحلل رحلات Uber وفقًا لأيام الأسبوع:

```
sns.distplot(data["Weekday"])
```



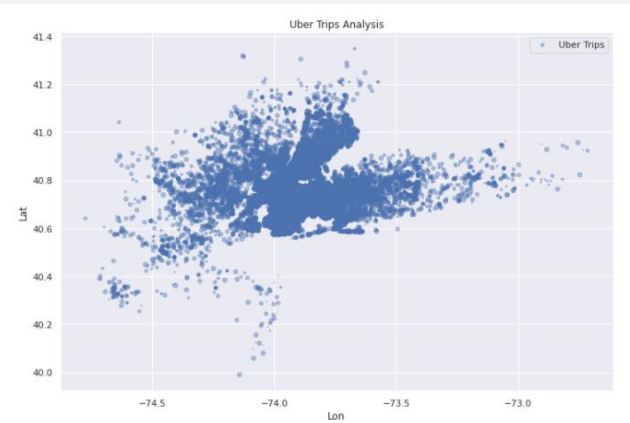
في الشكل 0 أعلاه يشير إلى يوم الأحد، وفي أيام الأحد، تكون رحلات Uber وأكثر من أيام السبت، لذا يمكننا القول إن الأشخاص يستخدمون Uber أيضًا للتنزه بدلاً من الذهاب إلى العمل فقط.في أيام السبت، تكون رحلات Uber هي الأدنى وفي يوم الاثنين هي الأعلى. دعنا الآن نلقي نظرة على الارتباط (correlation) بين الساعات وأيام الأسبوعفي رحلات Uber:

```
# Correlation of Weekday and Hour
df = data.groupby(["Weekday", "Hour"]).apply(lambda x: len(x))
df = df.unstack()
sns.heatmap(df, annot=False)
```



نظرًا لأننا نمتلك بيانات حول خطوط الطول والعرض، فيمكننا أيضًا رسم كثافة رحلات Uber وفقًا لمناطق مدينتك الجديدة:

```
data.plot(kind='scatter', x='Lon', y='Lat', alpha=0.4,
s=data['Day'], label='Uber Trips',
figsize=(12, 8), cmap=plt.get_cmap('jet'))
plt.title("Uber Trips Analysis")
plt.legend()
plt.show()
```



الملخص

هذه هي الطريقة التي يمكننا بها تحليل رحلات Uber لمدينة نيويورك. بعض الاستنتاجات التي حصلت عليها من هذا التحليل هي:

1. يوم الاثنين هو اليوم الأكثر ربحية لشركة Uber.
2. في أيام السبت، يستخدم عدد أقل من الأشخاص Uber.
3. 6 مساءً هو أكثر الأيام ازدحامًا بالنسبة لـ Uber.

4. في المتوسط، تبدأ زيادة رحلات Uber في حوالي الساعة 5 صباحًا.

5. تبدأ معظم رحلات Uber بالقرب من منطقة مانهاتن في نيويورك.

آمل أن تكون قد أحببت هذه المقالة حول تحليل رحلات Uber باستخدام بايثون.

35) تحليل بحث Google باستخدام بايثون Google Search Analysis with Python

يتم إجراء ما يقرب من 3.5 مليار عملية بحث على Google يوميًا، مما يعني أنه يتم إجراء ما يقرب من 40.000 عملية بحث كل ثانية على Google. لذا يعد بحث Google حالة استخدام رائعة لتحليل البيانات بناءً على استعلامات البحث. مع وضع ذلك في الاعتبار، في هذه المقالة، سوف أطلعك على مهمة تحليل بحث Google باستخدام بايثون.

تحليل بحث Google باستخدام بايثون

لا تمنح Google الكثير من الوصول إلى البيانات المتعلقة بطلبات البحث اليومية، ولكن يمكن استخدام تطبيق آخر من Google يُعرف باسم Google Trends لمهمة تحليل بحث Google. توفر Google Trends واجهة برمجة تطبيقات API يمكن استخدامها لتحليل عمليات البحث اليومية على Google. تُعرف واجهة برمجة التطبيقات هذه باسم pytrends، ويمكنك تثبيتها بسهولة في أنظمتك باستخدام الأمر pip:

```
pip install pytrends
```

أتمنى أن تكون قد قمت الآن بتثبيت مكتبة pytrends في أنظمتك بسهولة، فلنبدأ الآن بمهمة تحليل بحث Google عن طريق استيراد مكتبات بايثون الضرورية:

```
import pandas as pd
from pytrends.request import TrendReq
import matplotlib.pyplot as plt
trends = TrendReq()
```

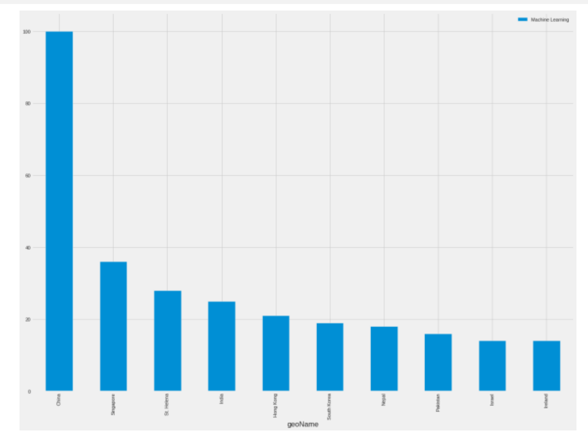
سأقوم هنا بتحليل مؤشرات بحث Google على الاستعلامات القائمة على جملة البحث "Machine Learning"، لذا فلنقم بإنشاء إطار بيانات لأهم 10 بلدان تبحث عن "Machine Learning" على Google:

```
trends.build_payload(kw_list=["Machine Learning"])
data = trends.interest_by_region()
data = data.sort_values(by="Machine Learning",
ascending=False)
data = data.head(10)
print(data)
```

```
               Machine Learning
geoName
China                       100
Singapore                    36
St. Helena                   28
India                        25
Hong Kong                    21
South Korea                  19
Nepal                        18
Pakistan                     16
Israel                       14
Ireland                      14
```

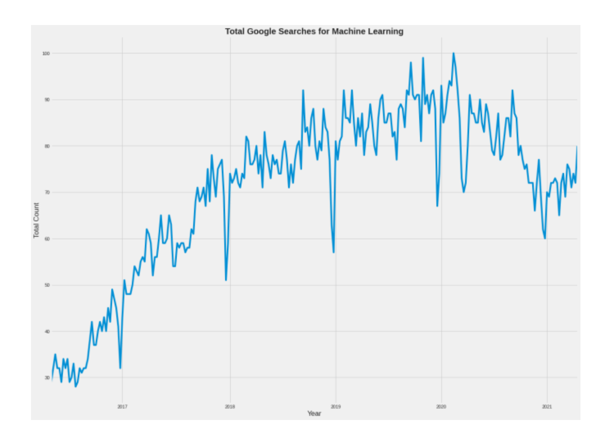
لذلك، وفقًا للنتائج المذكورة أعلاه، تتم معظم طلبات البحث المستندة إلى " Machine Learning" في الصين. يمكننا أيضًا رسم هذه البيانات باستخدام مخطط شريطي (bar chart):

```
data.reset_index().plot(x="geoName", y="Machine Learning",
                        figsize=(20,15), kind="bar")
plt.style.use('fivethirtyeight')
plt.show()
```



نظرًا لأننا نعلم جميعًا أن "Machine Learning" كان محط اهتمام العديد من الشركات والطلاب على مدار السنوات الثلاث أو الأربع الماضية، فلنلق نظرة على اتجاه عمليات البحث لمعرفة كيف زاد إجمالي استعلامات البحث المستندة إلى "Machine Learning " أو انخفض على Google:

```
data = TrendReq(hl='en-US', tz=360)
data.build_payload(kw_list=['Machine Learning'])
data = data.interest_over_time()
fig, ax = plt.subplots(figsize=(20, 15))
data['Machine Learning'].plot()
plt.style.use('fivethirtyeight')
plt.title('Total Google Searches for Machine Learning',
fontweight='bold')
plt.xlabel('Year')
plt.ylabel('Total Count')
plt.show()
```



الملخص

لذلك يمكننا أن نرى أن عمليات البحث القائمة على "Machine Learning" على Google بدأتفي الزيادةفي عام 2017 وتم إجراء أعلى عمليات البحثفي عام 2020 حتى اليوم. هذه هي الطريقة التي يمكننا بها تحليل عمليات بحث Google بناءً على أي كلمة رئيسية. يمكن لأي نشاط تجاري إجراء تحليل بحث Google لفهم ما يبحث عنه الأشخاص على Googleفي أي وقت. أتمنى أن تكون هذه المقالة قد أحببت تحليل بحث Google باستخدام بايثون.

36) تحليل الموازنة المالية مع بايثون Financial Budget Analysis with Python

لكل دولة موازنة مالية (financial budget) تصف قدرة الإنفاق الحكومية في مختلف قطاعات الاقتصاد. في هذه المقالة، سوف أطلعك على مهمة تحليل الموازنة المالية باستخدام بايثون.

ما هي الموازنة المالية؟

يوجد اليوم العديد من محللي البيانات الذين يأتون من خلفية غير برمجية. إذا كنت من خلفية تجارية، فقد تعرف ما هي الميزانية المالية. باختصار، إنه تقرير مفصل عن دخل ونفقات الحكومة عن سنة مالية.

قد تحصل على مهمة تحليل الموازنة المالية لبلد ما كل عام إذا كنت تعمل كمحلل بيانات في مجال الإعلام والاتصالات، حيث يتعين على وسائل الإعلام شرح أولويات الحكومة للسنة المالية الكاملة. في القسم أدناه، سوف آخذك من خلال برنامج تعليمي حول كيفية أداء مهمة تحليل الموازنة المالية باستخدام بايثون.

تحليل الموازنة المالية مع بايثون

أتمنى أن تكون قد فهمت الآن ما هي الموازنة المالية ومتى قد تحتاج إلى تحليلها كمحلل بيانات. دعونا نرى كيف يمكننا أداء مهمة تحليل الموازنة المالية باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد مكتبات بايثون الضرورية ومجموعة بيانات تحتوي على بيانات حول الموازنة المالية للهند لعام 2021.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("India_budget_2021.csv")
data.head()
```

	Department /Ministry	Fund allotted(in ₹crores)
0	MINISTRY OF AGRICULTURE	131531.19
1	DEPARTMENT OF ATOMIC ENERGY	18264.89
2	MINISTRY OF AYURVEDA, YOGA	2970.30
3	MINISTRY OF CHEMICALS AND FERTILISER	80714.94
4	MINISTRY OF CIVIL AVIATION	3224.67

دعونا نلقي نظرة على جميع الأقسام التي تغطيها هذه الموازنة:

```
print(data)
```

	Department /Ministry	Fund allotted(in ₹crores)
0	MINISTRY OF AGRICULTURE	131531.19
1	DEPARTMENT OF ATOMIC ENERGY	18264.89
2	MINISTRY OF AYURVEDA, YOGA	2970.30
3	MINISTRY OF CHEMICALS AND FERTILISER	80714.94
4	MINISTRY OF CIVIL AVIATION	3224.67
5	MINISTRY OF COAL	534.88
6	MINISTRY OF COMMERCE AND INDUSTRY	12768.25
7	MINISTRY OF COMMUNICATION	75265.22
8	MINISTRY OF CONSUMER AFFAIRS	256948.40
9	MINISTRY OF CORPORATE AFFAIRS	712.13
10	MINISTRY OF CULTURE	2687.99
11	MINISTRY OF DEFENCE	478195.62
12	MINISTRY OF DEVELOPMENT OF NORTH EASTERN REGION	2658.00
13	MINISTRY OF EARTH SCIENCES	1897.13
14	MINISTRY OF EDUCATION	93224.31
15	MINISTRY OF ELECTRONICS AND INFORMATION TECHNO...	9720.66
16	MINISTRY OF ENVIRONMENT, FOREST	2869.93
17	MINISTRY OF EXTERNAL AFFAIRS	18154.73
18	MINISTRY OF FINANCE	1386273.30
19	MINISTRY OF FISHERIES, ANIMAL HUSBANDRY	4322.82
20	MINISTRY OF FOOD PROCESSING INDUSTRIES	1308.66
21	MINISTRY OF HEALTH AND FAMILY WELFARE	73931.77
22	MINISTRY OF HEAVY INDUSTRIES	1017.08
23	MINISTRY OF HOME AFFAIRS	166546.94
24	MINISTRY OF HOUSING AND URBAN AFFAIRS	54581.00
25	MINISTRY OF INFORMATION AND BROADCASTING	4071.23
26	MINISTRY OF JAL SHAKTI	69053.02
27	MINISTRY OF LABOUR AND EMPLOYMENT	13306.50
28	MINISTRY OF LAW AND JUSTICE	3229.94
29	MINISTRY OF MICRO, SMALL AND MEDIUM ENTERPRISES	15699.65
30	MINISTRY OF MINES	1466.82
31	MINISTRY OF MINORITY AFFAIR	4810.77
32	MINISTRY OF NEW AND RENEWABLE ENERGY	5753.00
33	MINISTRY OF PANCHAYATI RAJ	913.43
34	MINISTRY OF PARLIAMENTARY AFFAIRS	65.07
35	MINISTRY OF PERSONNEL, PUBLIC GRIEVANCES	2097.24
36	MINISTRY OF PETROLEUM AND NATURAL GAS	15943.78
37	MINISTRY OF PLANNING	1062.77
38	MINISTRY OF PORTS, SHIPPING	1702.35
39	MINISTRY OF POWER	15322.00
40	THE PRESIDENT, PARLIAMENT, UNION PUBLIC SERVIC...	1687.57
41	MINISTRY OF RAILWAYS	110054.64
42	MINISTRY OF ROAD TRANSPORT AND HIGHWAY	118101.00
43	MINISTRY OF RURAL DEVELOPMENT	133689.50
44	MINISTRY OF SCIENCE AND TECHNOLOGY	14794.03
45	MINISTRY OF SKILL DEVELOPMENT	2785.23
46	MINISTRY OF SOCIAL JUSTICE AND EMPOWERMENT	11689.39
47	DEPARMENT OF SPACE	13949.09
48	MINISTRY OF STATISTICS	1409.13
49	MINISTRY OF STEEL	39.25
50	MINISTRY OF TEXTILES	3631.64
51	MINISTRY OF TOURISM	2026.77
52	MINISTRY OF TRIBAL AFFAIRS	7524.87
53	MINISTRY OF WOMEN AND CHILD DEVELOPMENT	24435.00
54	MINISTRY OF YOUTH AFFAIRS AND SPORTS	2596.14
55	NaN	NaN
56	GRAND TOTAL	3483235.63

يمكنني رؤية القيم المفقودة (NaN)في مجموعة البيانات هذه، دعنا نزيل قيم NaN ونتابع مهمة تحليل الميزانية المالية باستخدام بايثون:

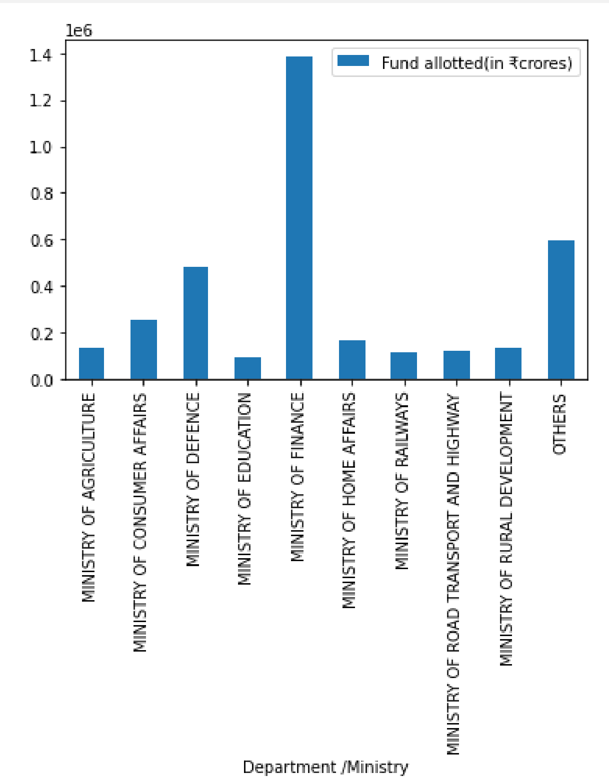
```
data.dropna()
```

أستطيع أن أرى أنه ليست كل الأقسام التي تغطيها مجموعة البيانات هذه هي الأقسام الرئيسية، حيث يمكن تغطية بعض الأقسامفي فئة أخرى. لذلك دعونا نجهز البيانات عن طريق اختيار الأقسام الرئيسية فقط ووضع جميع الأقسام الأخرىفي الفئة الأخرى:

	Department /Ministry	Fund allotted(in ₹crores)
0	MINISTRY OF AGRICULTURE	131531.19
1	MINISTRY OF CONSUMER AFFAIRS	256948.40
2	MINISTRY OF DEFENCE	478195.62
3	MINISTRY OF EDUCATION	93224.31
4	MINISTRY OF FINANCE	1386273.30
5	MINISTRY OF HOME AFFAIRS	166546.94
6	MINISTRY OF RAILWAYS	110054.64
7	MINISTRY OF ROAD TRANSPORT AND HIGHWAY	118101.00
8	MINISTRY OF RURAL DEVELOPMENT	133689.50
9	OTHERS	592971.08

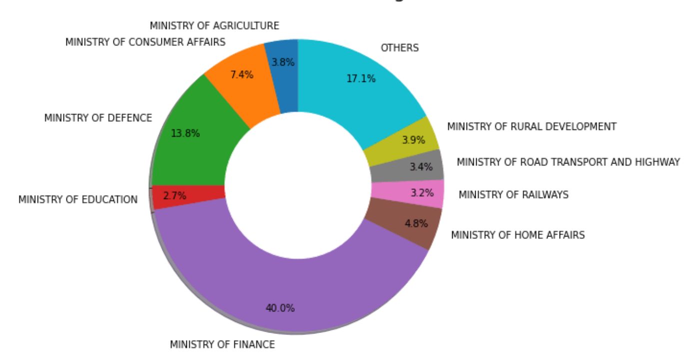
دعنا الآن نرسم هذه البيانات لإلقاء نظرة على أولويات الحكومة للسنة المالية:

```
data.plot.bar(x='Department /Ministry', y='Fund allotted(in ₹crores)')
```



يمكننا أن نرى أن قسم المالية (finance department) تحصل على أكبر حصة من الميزانية الإجمالية للحكومة. دعنا الآن نرسم هذه البيانات في مخطط دونات (donut plot) للحصول على رؤية واضحة لتوزيع الأموال بين جميع الأقسام:

```
df = data["Fund allotted(in ₹crores)"]
labels = data["Department /Ministry"]
plt.figure(figsize=(7,7))
plt.pie(df, labels=labels, autopct='%1.1f%%', startangle=90,
pctdistance=0.85, shadow =True)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Distribution of The Budget", fontsize=20)
plt.show()
```



الملخص

يمكننا أن نرى أن قسم المالية تحصل على 40٪ من الأموال. هذه هي الطريقة التي يمكننا بها تحليل مجموعة البيانات التي تحتوي على بيانات حول إيرادات ونفقات الحكومة لسنة مالية.

آمل أن تكون قد أحببت هذا المقال حول تحليل الميزانية المالية باستخدام بايثون.

37) تحليل أفضل خدمات البث باستخدام بايثون Best Streaming Service Analysis with Python

هناك الكثير من المنافسة بين جميع خدمات البث الرئيسية مثل Netflix و Prime Video و Hulu و +Disney. بصفتك عالم بيانات، قد تكون مهمة رائعة جداً بالنسبة لك أن تجد أفضل خدمة بث (Streaming Service) من بينها.في هذه المقالة، سأقدم لك مشروع علم البيانات حول أفضل تحليل لخدمة البث باستخدام بايثون.

تحليل أفضل خدمات البث

لتحليل أفضل خدمة بث، سأستخدم تقييمات العروض على جميع المنصات الرئيسية مثل Netflix و Prime Video و Hulu و +Disney.

تحتوي مجموعة البيانات التي سأستخدمها لمهمة تحليل أفضل خدمة بث على قائمة شاملة بجميع العروض التلفزيونية المتوفرة على الأنظمة الأساسية الأربعة التي نقوم بمقارنتهافي هذه المهمة.

أنا أستخدم مجموعة البيانات هذه للعثور على أفضل خدمة بث ولكن كمبتدئ ، يمكنك أيضًا استخدام مجموعة البيانات هذه لمهام مثل:

1. تحليل منصات البث.
2. تحليل تصنيفات IMBD و Rotten Tomatoes لجميع العروض.
3. تحليل الفئة العمرية المستهدفة لمعظم المسلسلات التليفزيونية..

تحليل أفضل خدمات البث باستخدام بايثون

لنبدأ الآن بمهمة تحليل أفضل خدمة بث باستخدام بايثون. سأبدأ هذه المهمة عن طريق استيراد جميع المكتبات ومجموعة البيانات الضرورية:

```
import numpy as np # linear algebra
import pandas as pd # data processing

import plotly
import plotly.express as px
from plotly.subplots import make_subplots
import seaborn as sns
import matplotlib.pyplot as plt

tv_shows = pd.read_csv('tv_shows.csv')
tv_shows.head()
```

```
tv_shows = pd.read_csv('tv_shows.csv')
tv_shows.head()
```

	Unnamed: 0	Title	Year	Age	IMDb	Rotten Tomatoes	Netflix	Hulu	Prime Video	Disney+	type
0	0	Breaking Bad	2008	18+	9.5	96%	1	0	0	0	1
1	1	Stranger Things	2016	16+	8.8	93%	1	0	0	0	1
2	2	Money Heist	2017	18+	8.4	91%	1	0	0	0	1
3	3	Sherlock	2010	16+	9.1	78%	1	0	0	0	1
4	4	Better Call Saul	2015	18+	8.7	97%	1	0	0	0	1

نظرًا لأننا نحلل البيانات فقط، فلن نحتاج إلى استخدام خوارزميات التعلم الآلي هنا. يمكن إنجاز معظم العمل من خلال رسم وتحليل تقييمات العروض على منصات البث.

تحضير البيانات

دعنا نجهز مجموعة البيانات حتى نتمكن من تحليل البيانات بسهولة. سأبدأفي إعداد البيانات بإسقاط القيم المكررة بناءً على عنوان العروض (title of the shows):

```
tv_shows.drop_duplicates(subset='Title',
          keep='first',inplace=True)
```

الآن،في قسم الكود أدناه، سأقوم بملء القيم المفقودةفي البيانات بالأصفار ثم تحويلها إلى أنواع بيانات عدد صحيح:

```
tv_shows['Rotten Tomatoes'] = tv_shows['Rotten
Tomatoes'].fillna('0%')
tv_shows['Rotten Tomatoes'] = tv_shows['Rotten
Tomatoes'].apply(lambda x : x.rstrip('%'))
tv_shows['Rotten Tomatoes'] = pd.to_numeric(tv_shows['Rotten
Tomatoes'])

tv_shows['IMDb'] = tv_shows['IMDb'].fillna(0)
tv_shows['IMDb'] = tv_shows['IMDb']*10
tv_shows['IMDb'] = tv_shows['IMDb'].astype('int')
```

سيكون رسم البيانات أمرًا سهلاً إذا حصلنا على 1 و 0 ثانيةفي الأعمدة المسماة Netflix و Hulu و Disney و Prime Video بتنسيق فئوي (categorical). قد يكون هناك احتمال أن يكون العرض نفسه متاحًافي أكثر من منصة واحدة:

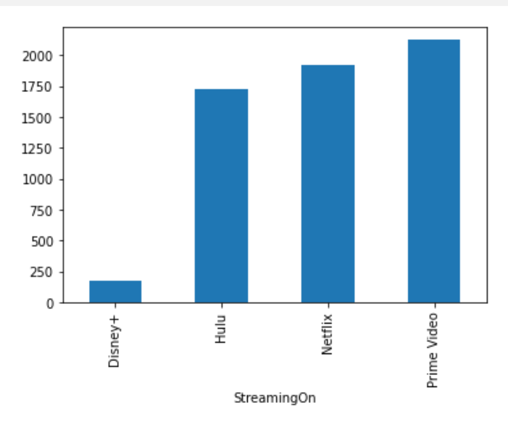
```
tv_shows_long=pd.melt(tv_shows[['Title','Netflix','Hulu','Disn
ey+',
                                    'Prime
Video']],id_vars=['Title'],
                       var_name='StreamingOn',
value_name='Present')
tv_shows_long = tv_shows_long[tv_shows_long['Present'] == 1]
tv_shows_long.drop(columns=['Present'],inplace=True)
```

سأقوم الآن بدمج هذه البيانات مع البيانات التي بدأنا بها ولكني سأقوم بإسقاط بعض الأعمدة غير المرغوب فيها:

```
tv_shows_combined = tv_shows_long.merge(tv_shows, on='Title',
how='inner')
tv_shows_combined.drop(columns = ['Unnamed: 0','Netflix',
                                  'Hulu', 'Prime Video',
'Disney+', 'type'], inplace=True)
```

الآن دعنا نرسم البيانات التي تكون فيها التقييمات أكثر من 1 لمعرفة كمية العروض التلفزيونية المتاحة على كل منصة:

```
tv_shows_both_ratings =
tv_shows_combined[(tv_shows_combined.IMDb > 0) &
tv_shows_combined['Rotten Tomatoes'] > 0]
tv_shows_combined.groupby('StreamingOn').Title.count().plot(ki
nd='bar')
```



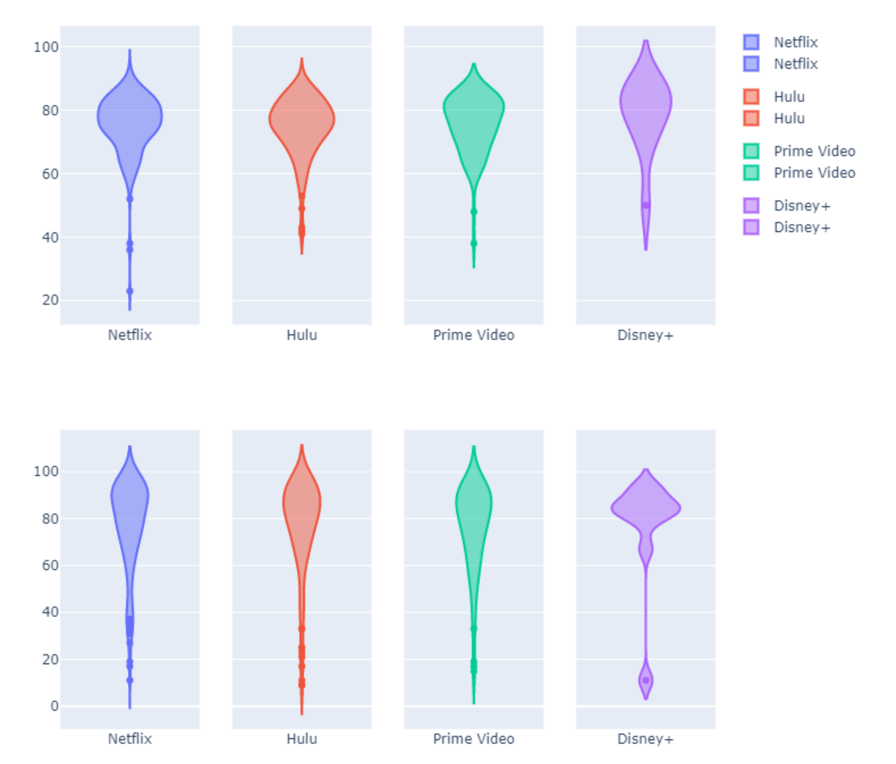
الخطوة النهائية: العثور على أفضل خدمة بث

الآن دعنا نرسم البيانات للعثور على أفضل خدمة بث بناءً على تقييماتهم. سأستخدم أولاً مخططات الكمان (violin charts) لقياس تقييمات المحتوى وحداثة منصة البث:

```
figure = []
figure.append(px.violin(tv_shows_both_ratings, x =
'StreamingOn', y = 'IMDb', color='StreamingOn'))
figure.append(px.violin(tv_shows_both_ratings, x =
'StreamingOn', y = 'Rotten Tomatoes', color='StreamingOn'))
fig = make_subplots(rows=2, cols=4, shared_yaxes=True)

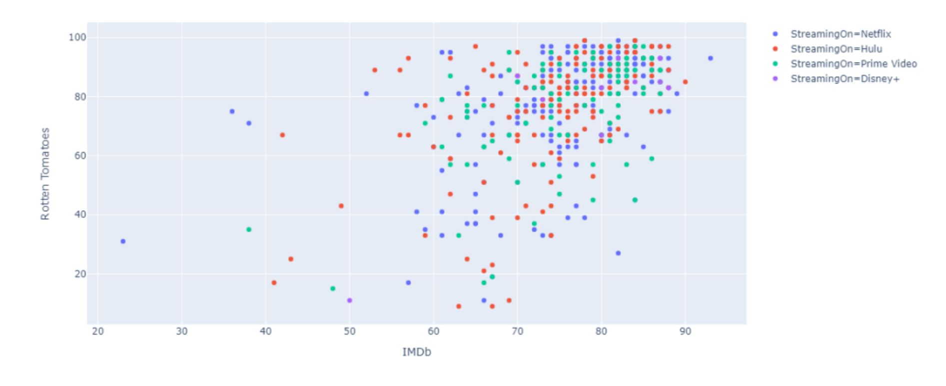
for i in range(2):
    for j in range(4):
        fig.add_trace(figure[i]['data'][j], row=i+1, col=j+1)

fig.update_layout(autosize=False, width=800, height=800)
fig.show()
```



الآن، دعنا نستخدم مخطط مبعثر (scatter plot) لمقارنة التقييمات بين IMDB و Rotten Tomatoes لمقارنة أي منصة بث لديها أفضل التقييمات في كل من منصات تصنيف المستخدم:

```
px.scatter(tv_shows_both_ratings, x='IMDb',
     y='Rotten Tomatoes',color='StreamingOn ')
```



الملخص

باستخدام مخطط الكمان يمكننا ملاحظة ما يلي:

1. تحتوي مقاطع فيديو Hulu و Netflix و Amazon على بيانات مهمة. مع زيادة المحتوى، تنخفض الجودة لجميع الثلاثة.
2. أصبح Prime Videos أكثر كثافةفي النصف العلوي عند النظر إلى IMDB ويعمل بشكل جيدفي الوضع البارد.
3. أصبحت +Disney جديدة، وقد حققت أيضًا نجاحًا كبيرًافي هذا المجال.

باستخدام مخطط التبعثر يمكننا أن نلاحظ أنه من الواضح تمامًا أن Amazon Prime يعمل جيدًافي الربع الرابع. حتى باستخدام المخطط الشريطي، يمكننا ملاحظة أن Amazon Prime كانت تحتوي على كمية كبيرة من المحتوى. لذا بالنظر إلى جميع منصات البث، يمكننا أن نستنتج أن Amazon Prime أفضل من حيث الجودة والكمية.

آمل أن تكون قد أحببت هذه المقالة حول مشروع علم البيانات حول تحليل أفضل خدمة البث باستخدام لغة برمجة بايثون.

38) تحليل معدل المواليد باستخدام بايثون Birth Rate Analysis with python

دعنا نلقي نظرة على البيانات المتاحة مجانًا عن الولادات في الولايات المتحدة، والتي تقدمها مراكز السيطرة على الأمراض (CDC). يمكن العثور على هذه البيانات في births.csv .

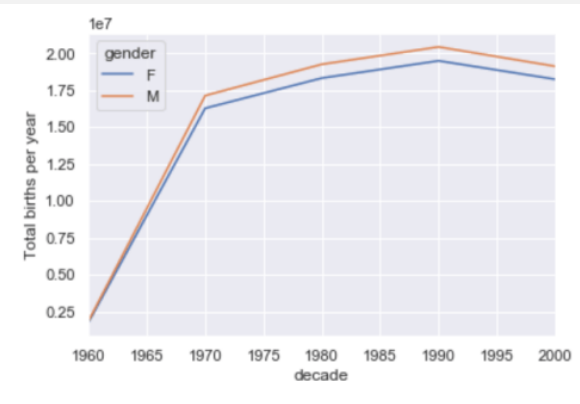
```
import pandas as pd
births = pd.read_csv("births.csv") print(births.head())
births['day'].fillna(0, inplace=True) births['day'] =
births['day'].astype(int)
```

```
   year  month  day gender  births
0  1969      1  1.0      F    4046
1  1969      1  1.0      M    4440
2  1969      1  2.0      F    4454
3  1969      1  2.0      M    4548
4  1969      1  3.0      F    4548
```

```
births['decade'] = 10 * (births['year'] // 10)
births.pivot_table('births', index='decade', columns='gender',
aggfunc='sum')
print(births.head())
```

نرى على الفور أن عدد المواليد الذكور يفوق عدد المواليد الإناث في كل عقد. لرؤية هذا الاتجاه بشكل أكثر وضوحًا، يمكننا استخدام أدوات الرسم المدمجة في Pandas لرسم العدد الإجمالي للمواليد حسب السنة:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
birth_decade = births.pivot_table('births', index='decade',
columns='gender', aggfunc='sum')
birth_decade.plot()
plt.ylabel("Total births per year")
plt.show()
```



المزيد من استكشاف البيانات

هناك بعض الميزات المثيرة للاهتمام التي يمكننا سحبها من مجموعة البيانات هذه باستخدام أدوات Pandas . يجب أن نبدأ بتنظيف البيانات قليلاً، وإزالة القيم المتطرفة (outliers) التي تسببها التواريخ المكتوبة بشكل خاطئ (mistyped dates) أو القيم المفقودة (missing values). إحدى الطرق السهلة لإزالة كل ذلك مرة واحدة هي قص القيم المتطرفة، وسنفعل ذلك من خلال عملية قص سيجما (sigma-clipping) القوية:

```
import numpy as np
quartiles = np.percentile(births['births'], [25, 50, 75])
mu = quartiles[1]
sig = 0.74 * (quartiles[2] - quartiles[0])
```

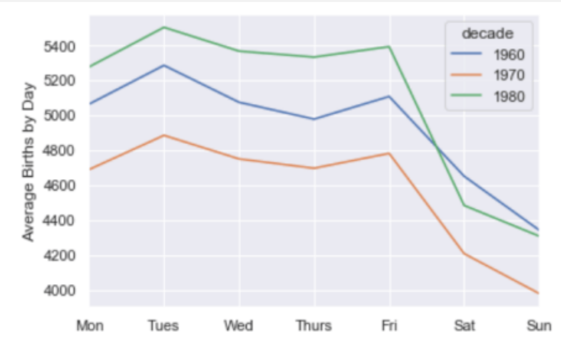
هذا السطر الأخير هو تقدير قوي لمتوسط العينة، حيث يأتي 0.74 من النطاق الربعي لتوزيع غاوسي. باستخدام هذا، يمكننا استخدام التابع query() لتصفية الصفوف التي تحتوي على ولادات خارج هذه القيم:

```
births = births.query('(births > @mu - 5 * @sig) & (births <
@mu + 5 * @sig)')
births['day'] = births['day'].astype(int)
births.index = pd.to_datetime(10000 * births.year+
 *100              births.month+
              births.day, format='%Y%m%d ')

births['dayofweek'] = births.index.dayofweek
```

باستخدام هذا يمكننا رسم المواليدفي أيام الأسبوع لعدة عقود:

```
births.pivot_table('births', index='dayofweek,'
        columns='decade', aggfunc='mean').plot()
plt.gca().set_xticklabels(['Mon', 'Tues', 'Wed', 'Thurs',
'Fri', 'Sat', 'Sun'])
plt.ylabel('mean births by day');
plt.show()
```



يبدو أن الولادات أقل شيوعًافي عطلات نهاية الأسبوع منهافي أيام الأسبوع! لاحظ أن التسعينيات والألفينيات من القرن الماضي مفقودة لأن بيانات مركز السيطرة على الأمراض تحتوي فقط على شهر الميلاد بدءًا من عام 1989.

وجهة نظر أخرى مثيرة للاهتمام هي رسم متوسط عدد المواليد حسب اليوم من السنة. دعنا أولاً نجمع البيانات حسب الشهر واليوم بشكل منفصل:

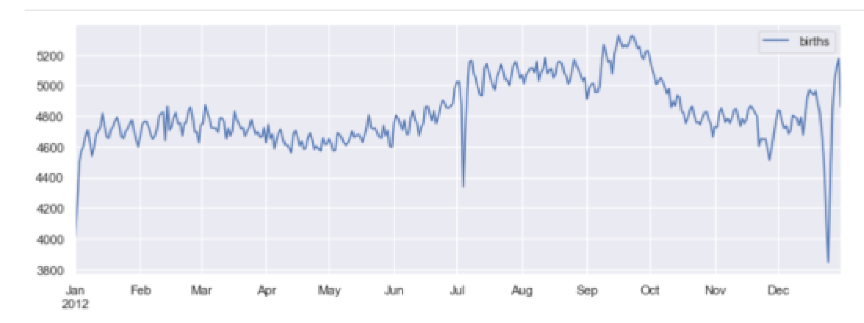
```
births_month = births.pivot_table('births',
[births.index.month, births.index.day])
print(births_month.head())

births_month.index = [pd.datetime(2012, month, day )
          for (month, day) in births_month.index]
print(births_month.head())
```

```
        births
1 1   4009.225
  2   4247.400
  3   4500.900
  4   4571.350
  5   4603.625
              births
2012-01-01  4009.225
2012-01-02  4247.400
2012-01-03  4500.900
2012-01-04  4571.350
2012-01-05  4603.625
```

بالتركيز على الشهر واليوم فقط، لدينا الآن سلسلة زمنية تعكس متوسط عدد المواليد حسب تاريخ السنة. من هذا، يمكننا استخدام طريقة plot لرسم البيانات. يكشف عن بعض الاتجاهات المثيرة للاهتمام:

```
fig, ax = plt.subplots(figsize=(12, 4))
births_month.plot(ax=ax)
plt.show()
```



39) تحليل المشاعر في تقييمات منتجات Amazon باستخدام بايثون Amazon Product Reviews Sentiment Analysis with Python

Amazon هي شركة أمريكية متعددة الجنسيات تركز على التجارة الإلكترونية والحوسبة السحابية والبث الرقمي ومنتجات الذكاء الاصطناعي. لكنها معروفة بشكل أساسي بمنصة التجارة الإلكترونية الخاصة بها والتي تعد واحدة من أكبر منصات التسوق عبر الإنترنت اليوم. هناك الكثير من العملاء الذين يشترون المنتجات من Amazon حتى أن Amazon تكسب اليوم في المتوسط 638.1 مليون دولار في اليوم. لذا، فإن وجود مثل هذه القاعدة الكبيرة من العملاء، سيصبح مشروعًا رائعًا لعلوم البيانات إذا تمكنا من تحليل مشاعر مراجعات منتجات Amazon. لذلك، في هذه المقالة، سوف أطلعك على مهمة تحليل آراء مراجعات منتجات Amazon باستخدام بايثون.

تحليل المشاعر في تقييمات منتجات Amazon باستخدام بايثون

تم تنزيل مجموعة البيانات التي أستخدمها لمهمة تحليل آراء مراجعات منتجات Amazon من Kaggle. تحتوي مجموعة البيانات هذه على مراجعات المنتجات لأكثر من 568000 عميل اشتروا منتجات من Amazon. فلنبدأ هذه المهمة عن طريق استيراد مكتبات بايثون ومجموعة البيانات الضرورية:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data = pd.read_csv("Reviews.csv")
print(data.head())
```

```
   Id   ProductId          UserId                      ProfileName  \
0   1  B001E4KFG0  A3SGXH7AUHU8GW                       delmartian
1   2  B00813GRG4  A1D87F6ZCVE5NK                           dll pa
2   3  B000LQOCH0   ABXLMWJIXXAIN  Natalia Corres "Natalia Corres"
3   4  B000UA0QIQ  A395BORC6FGVXV                             Karl
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T    Michael D. Bigham "M. Wassir"

   HelpfulnessNumerator  HelpfulnessDenominator  Score        Time  \
0                     1                       1      5  1303862400
1                     0                       0      1  1346976000
2                     1                       1      4  1219017600
3                     3                       3      2  1307923200
4                     0                       0      5  1350777600

                 Summary                                               Text
0  Good Quality Dog Food  I have bought several of the Vitality canned d...
1      Not as Advertised  Product arrived labeled as Jumbo Salted Peanut...
2  "Delight" says it all  This is a confection that has been around a fe...
3         Cough Medicine  If you are looking for the secret ingredient i...
4            Great taffy  Great taffy at a great price.  There was a wid...
```

قبل المضي قدمًا، دعنا نلقي نظرة على بعض المعلومات المطلوبة من مجموعة البيانات هذه:

```
print(data.describe())
```

```
                  Id  HelpfulnessNumerator  HelpfulnessDenominator  \
count  568454.000000         568454.000000            568454.00000
mean   284227.500000              1.743817                 2.22881
std    164098.679298              7.636513                 8.28974
min         1.000000              0.000000                 0.00000
25%    142114.250000              0.000000                 0.00000
50%    284227.500000              0.000000                 1.00000
75%    426340.750000              2.000000                 2.00000
max    568454.000000            866.000000               923.00000

               Score          Time
count  568454.000000  5.684540e+05
mean        4.183199  1.296257e+09
std         1.310436  4.804331e+07
min         1.000000  9.393408e+08
25%         4.000000  1.271290e+09
50%         5.000000  1.311120e+09
75%         5.000000  1.332720e+09
max         5.000000  1.351210e+09
```

نظرًا لأن مجموعة البيانات هذه كبيرة جدًا، فهي تحتوي على بعض القيم المفقودة (missing values)، لذا دعنا نزيل جميع الصفوف التي تحتوي على القيم المفقودة:

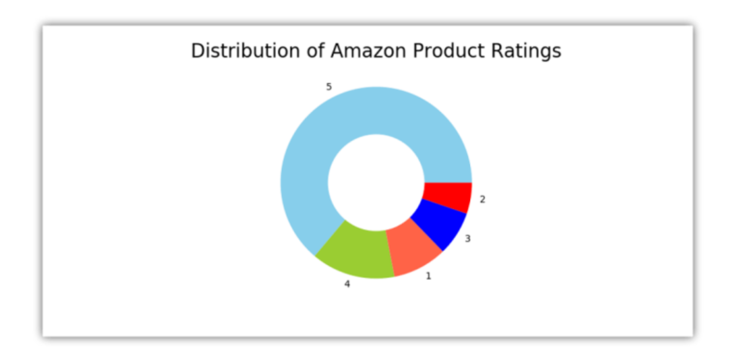
```
data = data.dropna()
```

تحليل المشاعر لمراجعات منتجات Amazon

يحتوي عمود النقاط (Score column) في مجموعة البيانات هذه على التصنيفات التي منحها العملاء للمنتج بناءً على تجربتهم مع المنتج. لذلك دعونا نلقي نظرة على تفاصيل التصنيف لمعرفة كيف يقوم معظم العملاء بتقييم المنتجات التي يشترونها من Amazon :

```
ratings = data["Score"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(10, 8))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Distribution of Amazon Product Ratings",
fontsize=20)
plt.show()
```



وفقًا للشكل أعلاه، صنف أكثر من نصف الأشخاص المنتجات التي اشتروها من Amazon بـ 5 نجوم، وهو أمر جيد. الآن، سأضيف ثلاثة أعمدة أخرى إلى مجموعة البيانات هذه على أنها موجبة (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر(sentiment scores) لمراجعات العملاء المذكورة في عمود النص (Text column) لمجموعة البيانات:

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Text"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Text"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Text"]]
print(data.head())
```

```
   Id   ProductId          UserId  ... Positive  Negative  Neutral
0   1  B001E4KFG0  A3SGXH7AUHU8GW  ...    0.305     0.000    0.695
1   2  B00813GRG4  A1D87F6ZCVE5NK  ...    0.000     0.138    0.862
2   3  B000LQOCH0   ABXLMWJIXXAIN  ...    0.155     0.091    0.754
3   4  B000UA0QIQ  A395BORC6FGVXV  ...    0.000     0.000    1.000
4   5  B006K2ZZ7K  A1UQRSCLF8GW1T  ...    0.448     0.000    0.552

[5 rows x 13 columns]
```

دعنا الآن نرى كيف قام معظم الناس بتقييم المنتجات التي اشتروها من أمازون:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
```

```
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

لذلك، يكون معظم الأشخاص محايدين عند إرسال تجاربهم مع المنتجات التي اشتروها من Amazon. الآن دعنا نرى إجمالي جميع درجات المشاعر :

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  109328.1269999992
Negative:  24033.022999999564
Neutral:  435043.95799998916
```

لذلك يمكننا القول إن معظم المراجعات للمنتجات المتاحة على Amazon إيجابية، حيث أن إجمالي درجات المشاعر الإيجابية والمحايدة أعلى بكثير من النتائج السلبية.

الملخص

إذن هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات المنتجفي Amazon. هناك الكثير من العملاء الذين يشترون المنتجات من Amazon حتى أن Amazon تكسب اليومفي المتوسط 638.1 مليون دولارفي اليوم. لذا، فإن وجود مثل هذه القاعدة الكبيرة من العملاء، سيصبح مشروعًا رائعًا لعلوم البيانات إذا تمكنا من تحليل مشاعر مراجعات منتجات Amazon. آمل أن تكون قد أحببت هذه المقالة حول تحليل المشاعر لمراجعات Amazon باستخدام بايثون.

40) تحليل مشاعر تقييمات الفندق مع بايثون Hotel Reviews Sentiment Analysis with Python

عندما نبحث عن فنادق لقضاء الإجازة أو السفر، نفضل دائمًا فندقًا معروفًا بخدماته. أفضل طريقة لمعرفة ما إذا كان الفندق مناسبًا لك أم لا هي معرفة ما يقوله الناس عن الفندق الذي أقام هناك من قبل. من الصعب للغاية الآن قراءة تجربة كل شخص أبدى رأيه في خدمات الفندق. هذا هو المكان الذي تأتي فيه مهمة تحليل المشاعر (sentiment analysis).في هذه المقالة، سوف أطلعك على مهمة تحليل مشاعر تقييمات الفندق (Hotel Reviews Sentiment Analysis) باستخدام بايثون.

تحليل مشاعر تقييمات الفندق مع بايثون

يتم جمع مجموعة البيانات التي أستخدمها لمهمة تحليل آراء الفنادق من Kaggle. يحتوي على بيانات حول 20000 تقييم للأشخاص حول خدمات الفنادق التي أقاموا فيها لقضاء عطلة أو رحلة عمل أو أي نوع من الرحلات. تحتوي مجموعة البيانات هذه على عمودين فقط كمراجعات (Reviews) وتقييمات (Ratings) للعملاء. لذا فلنبدأ بمهمة تحليل آراء الفنادق باستخدام بايثون من خلال استيراد مكتبات بايثون ومجموعة البيانات اللازمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data = pd.read_csv("hotel_reviews.csv")
print(data.head())
```

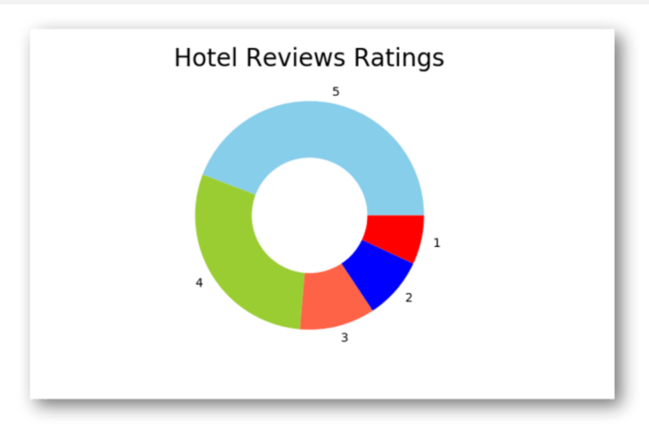
	Review	Rating
0	nice hotel expensive parking got good deal sta...	4
1	ok nothing special charge diamond member hilto...	2
2	nice rooms not 4* experience hotel monaco seat...	3
3	unique, great stay, wonderful time hotel monac...	5
4	great stay great stay, went seahawk game aweso...	5

مجموعة البيانات هذه كبيرة جدًا ولحسن الحظ لا توجد قيم مفقودة، لذا دون إضاعة أي وقت، دعنا نلقي نظرة سريعة على توزيع تقييمات العملاء:

```
ratings = data["Rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
```

```
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Hotel Reviews Ratings", fontsize=20)
plt.show()
```



يمكن ملاحظة أن معظم النزلاء قيموا الخدمات الفندقية بـ 5 نجوم و4 نجوم. لذلك وفقًا للتصنيفات المذكورة أعلاه، يمكننا القول إن معظم الضيوف راضون عن خدمات الفندق الذي أقاموا فيه. دعنا الآن نمضي قدمًا من خلال تحليل مشاعر تقييمات الفنادق. لتحليل وجهة نظر تقييمات الفندق، سأضيف ثلاثة أعمدة إضافية إلى مجموعة البيانات هذه على أنها إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral)من خلال حساب درجات المشاعر للتقييمات (sentiment scores):

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["Review"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["Review"]]
print(data.head())
```

Review	Rating	Positive	Positive	Negative	Neutral
nice hotel expensive parking got good deal sta...	4	0.285	0.285	0.072	0.643
ok nothing special charge diamond member hilto...	2	0.189	0.189	0.110	0.701
nice rooms not 4* experience hotel monaco seat...	3	0.219	0.219	0.081	0.700
unique, great stay, wonderful time hotel monac...	5	0.385	0.385	0.060	0.555
great stay great stay, went seahawk game aweso...	5	0.221	0.221	0.135	0.643

وفقًا للتقييمات (reviews)، يبدو أن ضيوف الفندق راضون عن الخدمات، فلنلقِ الآن نظرة على رأي معظم الناس في خدمات الفنادق بناءً على مشاعر تقييمهم:

```
x = sum(data["Positive"])
```

```
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

وبالتالي، يشعر معظم الناس بالحياد (neutral) تجاه خدمات الفندق. الآن دعنا نلقي نظرة فاحصة على نتائج المشاعر:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  6359.91000000002
Negative:  1473.4750000000038
Neutral:  12657.627999999937
```

وبالتالي، وفقًا للنتائج المذكورة أعلاه، تم تصنيف أكثر من 12000 مراجعة على أنها محايدة، وتم تصنيف أكثر من 6000 مراجعة على أنها إيجابية. لذلك يمكن القول إن الناس سعداء حقًا بخدمات الفنادق التي أقاموا فيها حيث أن التقييمات السلبية أقل من 1500.

الملخص

هذه هي الطريقة التي يمكنك بها تحليل مشاعر تقييمات الفنادق. أفضل طريقة لمعرفة ما إذا كان الفندق مناسبًا لك أم لا هي معرفة ما يقوله الناس عن الفندق الذي أقام هناك من قبل. هذا هو المكان الذي يمكن أن تساعدك فيه مهمة تحليل مشاعر تقييمات الفندق على تحديد ما إذا كان الفندق مناسبًا لرحلتك أم لا. آمل أن تكون قد أحببت هذه المقالة حول تحليل المشاعر لتقييمات الفنادق باستخدام بايثون.

41) تحليل المشاعر في متجر Google Play باستخدام بايثون Google Play Store Sentiment Analysis using Python

تحليل المشاعر (Sentiment analysis) هو تصنيف لمراجعات العميل أو تعليقاته على أنها إيجابية (positive) وسلبية (negative) وأحيانًا محايدة (neutral) أيضًا. تحلل معظم الأنشطة التجارية مشاعر عملائها حول منتجاتهم أو خدماتهم لمعرفة ما يريده عملاؤهم منهم. يحتوي متجر Google play على ملايين التطبيقات مع مراجعاتها، لذا ستكون حالة استخدام جيدة لتحليل المشاعر لتحليل مشاعر التطبيقات المتاحة على متجر Google play. لذلك،في هذه المقالة، سوف أطلعك على مهمة تحليل المشاعرفي متجر Google Play باستخدام بايثون.

تحليل المشاعر في متجر Google Play

يمكن العثور على متجر Google Play على جميع الهواتف الذكية والأجهزة اللوحية التي تعمل بنظام Android. هذا هو متجر تطبيقات Google الرسمي لنظام التشغيل Android. يحتوي على ملايين التطبيقات مع مراجعاتها حتى نتمكن من استخدام مثل هذا القدر من البيانات لأي مهمة تتعلق بعلوم البيانات. تحليل آراء العملاء وتعليقاتهم هو ما نقوم بهفي مهمة تحليل المشاعر. بعد قولي هذا،في القسم أدناه، سوف أطلعك على مهمة تحليل المشاعرفي متجر Google Play باستخدام بايثون. يمكن تنزيل مجموعة البيانات التي أستخدمهافي هذه المهمة من هنا.

تحليل المشاعر في متجر Google Play باستخدام بايثون

سأبدأ هذه المهمة بقراءة مجموعة البيانات. يتم تنزيل مجموعة البيانات التي أستخدمها هنا من Kaggle والتي تم جمعها من متجر Google Play. فلنبدأ هذه المهمة بقراءة مجموعة البيانات:

```
from itertools import count
from nltk.util import pr
import pandas as pd
data = pd.read_csv("user_reviews.csv")
print(data.head())
```

```
                  App  ... Sentiment_Subjectivity
0  10 Best Foods for You  ...               0.533333
1  10 Best Foods for You  ...               0.288462
2  10 Best Foods for You  ...                    NaN
3  10 Best Foods for You  ...               0.875000
4  10 Best Foods for You  ...               0.300000
```

قبل المضي قدمًا، دعنا نلقي نظرة على ما إذا كانت مجموعة البيانات هذه تحتوي على أي قيم مفقودة أم لا:

```
print(data.isnull().sum())
```

```
App                           0
Translated_Review         26868
Sentiment                 26863
Sentiment_Polarity        26863
Sentiment_Subjectivity    26863
dtype: int64
```

إذاً تحتوي على بعض القيم الفارغة، سأقوم بإنشاء مجموعة بيانات جديدة بإسقاط القيم الخالية:

```
data = data.dropna()
print(data.isnull().sum())
```

```
App                       0
Translated_Review         0
Sentiment                 0
Sentiment_Polarity        0
Sentiment_Subjectivity    0
dtype: int64
```

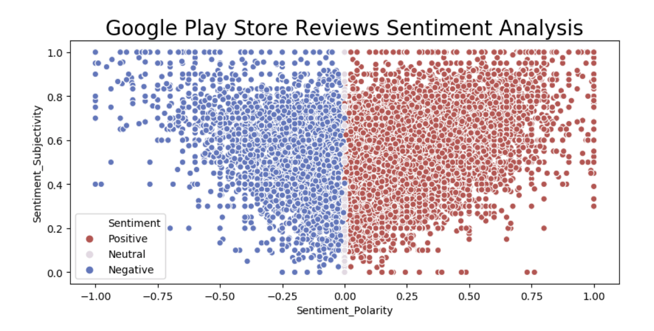
الآن لتحليل المشاعر الخاصة بمراجعات متجر google play ، سأضيف ثلاثة أعمدة جديدةفي مجموعة البيانات من خلال فهم مشاعر كل مراجعة للعميل على أنها إيجابية وسلبية ومحايدة:

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i 
in data["Translated_Review"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i 
in data["Translated_Review"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i 
in data["Translated_Review"]]
print(data.head())
```

```
                   App                                  Translated_Review  ... Negative  Neutral
0  10 Best Foods for You  I like eat delicious food. That's I'm cooking ...  ...      0.0    0.466
3  10 Best Foods for You       Works great especially going grocery store  ...      0.0    0.549
4  10 Best Foods for You                                      Best idea us  ...      0.0    0.323
5  10 Best Foods for You                                          Best way  ...      0.0    0.192
```

والآن كخطوة أخيرة، دعنا نلقي نظرة على مشاعر العملاء حول التطبيقات المتاحةفي متجر Google play باستخدام مخطط مبعثر (scatter plot):

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(15, 10))
sns.scatterplot(data['Sentiment_Polarity'], 
data['Sentiment_Subjectivity'], 
                hue = data['Sentiment'], edgecolor='white', 
palette="twilight_shifted_r")
plt.title("Google Play Store Reviews Sentiment Analysis", 
fontsize=20)
plt.show()
```



الملخص

إذن هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات متجر google play تحليل المشاعر هو تصنيف لمراجعات العميل أو تعليقاته على أنها إيجابية وسلبية ومحايدة. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة تحليل مشاعر متجر Google play باستخدام بايثون.

42) تحليل مشاعر مراجعات Amazon Alexa باستخدام بايثون Amazon Alexa Reviews Sentiment Analysis using Python

Amazon Alexa هي خدمة صوتية مستندة إلى السحابة تم تطويرها بواسطة Amazon تتيح للعملاء التفاعل مع التكنولوجيا. يوجد حاليًا أكثر من 40 مليون مستخدم لـ Alexa حول العالم، لذا فإن تحليل مشاعر المستخدمين حول Alexa سيكون مشروعًا جيداً لعلم البيانات. لذا، إذا كنت تريد معرفة كيفية تحليل مشاعر المستخدمين باستخدام Amazon Alexa، فهذه المقالة مناسبة لك.في هذه المقالة، سأوجهك خلال مهمة تحليل آراء Amazon Alexa باستخدام بايثون.

تحليل مشاعر مراجعات Amazon Alexa باستخدام بايثون

تم جمع مجموعة البيانات التي أستخدمها لمهمة تحليل المشاعر لمراجعات Amazon Alexa من Kaggle. يحتوي على بيانات حول التصنيفات بين 1 و 5 ، وتاريخ المراجعات ، وتعليقات العملاء حول تجربتهم مع Alexa. لذلك دعونا نستورد مجموعة بيانات ومكتبات بايثون الضرورية التي نحتاجها لهذه المهمة:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments = SentimentIntensityAnalyzer()

data =
pd.read_csv("https://raw.githubusercontent.com/amankharwal/Web
site-data/master/amazon_alexa.tsv", delimiter='\t')
print(data.head())
```

	rating	date	variation	verified_reviews	feedback
0	5	31-Jul-18	Charcoal Fabric	Love my Echo!	1
1	5	31-Jul-18	Charcoal Fabric	Loved it!	1
2	4	31-Jul-18	Walnut Finish	Sometimes while playing a game, you can answer...	1
3	5	31-Jul-18	Charcoal Fabric	I have had a lot of fun with this thing. My 4 ...	1
4	5	31-Jul-18	Charcoal Fabric	Music	1

لنبدأ بإلقاء نظرة على بعض المعلومات الموجودةفي تلك البيانات لمعرفة ما إذا كنا بحاجة إلى تغييرها أم لا:

```
print(data.describe())
print(data.isnull().sum())
print(data.columns)
```

```
            rating     feedback
count  3150.000000  3150.000000
mean      4.463175     0.918413
std       1.068506     0.273778
min       1.000000     0.000000
25%       4.000000     1.000000
50%       5.000000     1.000000
75%       5.000000     1.000000
max       5.000000     1.000000
rating              0
date                0
variation           0
verified_reviews    0
feedback            0
dtype: int64
Index(['rating', 'date', 'variation', 'verified_reviews', 'feedback'], dtype='object')
```

يحتوي عمود تصنيف مجموعة البيانات (dataset's rating column) على التقييمات التي قدمها مستخدمو Amazon Alexa بمقياس من 1 إلى 5، حيث يمثل الرقم 5 أفضل تقييم يمكن للمستخدم تقديمه. لذلك دعونا نلقي نظرة على توزيع التقييمات التي منحها مستخدمو Amazon Alexa:

```
ratings = data["rating"].value_counts()
numbers = ratings.index
quantity = ratings.values

custom_colors = ["skyblue", "yellowgreen", 'tomato', "blue",
"red"]
plt.figure(figsize=(5, 5))
plt.pie(quantity, labels=numbers, colors=custom_colors)
central_circle = plt.Circle((0, 0), 0.5, color='white')
fig = plt.gcf()
fig.gca().add_artist(central_circle)
plt.rc('font', size=12)
plt.title("Amazon Alexa Reviews", fontsize=20)
plt.show()
```



من الشكل أعلاه، يمكننا أن نرى أن معظم العملاء قد صنفوا Amazon Alexa بمافي ذلك جميع متغيراتها على أنها 5. لذا فهذا يعني أن معظم العملاء سعداء بخدمة Amazon Alexa .

تحليل مشاعر مراجعات Amazon Alexa

الآن دعنا ننتقل إلى مهمة تحليل المشاعر لمراجعات Alexa. يحتوي عمود المراجعات التي تم التحقق منها (verified_reviews column)في مجموعة البيانات على جميع المراجعات التي قدمها عملاء Amazon Alexa. لذلك دعونا نضيف أعمدة جديدة إلى هذه البيانات على أنها أعمدة إيجابية (Positive) وسلبية (Negative) ومحايدة (Neutral) من خلال حساب درجات المشاعر للمراجعات (sentiment scores):

```
sentiments = SentimentIntensityAnalyzer()
data["Positive"] = [sentiments.polarity_scores(i)["pos"] for i
in data["verified_reviews"]]
data["Negative"] = [sentiments.polarity_scores(i)["neg"] for i
in data["verified_reviews"]]
data["Neutral"] = [sentiments.polarity_scores(i)["neu"] for i
in data["verified_reviews"]]
print(data.head())
```

```
   rating       date         variation  ... Positive  Negative  Neutral
0       5  31-Jul-18  Charcoal Fabric   ...    0.692     0.000    0.308
1       5  31-Jul-18  Charcoal Fabric   ...    0.807     0.000    0.193
2       4  31-Jul-18    Walnut Finish   ...    0.114     0.102    0.784
3       5  31-Jul-18  Charcoal Fabric   ...    0.383     0.000    0.617
4       5  31-Jul-18  Charcoal Fabric   ...    0.000     0.000    1.000
```

دعنا الآن نلخص درجات المشاعر لكل عمود لفهم ما يعتقده معظم عملاء Amazon Alexa بشأنه:

```
x = sum(data["Positive"])
y = sum(data["Negative"])
z = sum(data["Neutral"])

def sentiment_score(a, b, c):
    if (a>b) and (a>c):
        print("Positive 😊 ")
    elif (b>a) and (b>c):
        print("Negative 😠 ")
    else:
        print("Neutral 🙂 ")
sentiment_score(x, y, z)
```

```
Neutral 🙂
```

وبالتالي، فإن الناتج النهائي الذي نحصل عليه يكون محايداً. هذا يعني أن معظم المستخدمين يشعرون بالحياد تجاه خدمات Amazon Alexa. دعنا الآن نرى مجموع درجات المشاعر لكل عمود:

```
print("Positive: ", x)
print("Negative: ", y)
print("Neutral: ", z)
```

```
Positive:  1035.4579999999983
Negative:  96.79999999999995
Neutral:  1936.740999999996
```

لذلك يمكننا أن نرى أن الايجابي والمحايد أعلى من 1000 حيث يكون السلبي أقل من 100. وهذا يعني أن معظم عملاء Amazon Alexa راضون عن خدماتها.

الملخص

هذه هي الطريقة التي يمكننا بها تحليل مشاعر مراجعات Amazon Alexa باستخدام لغة برمجة بايثون. يوجد حاليًا أكثر من 40 مليون مستخدم لـ Alexa حول العالم، لذا فإن تحليل مشاعر المستخدمين حول Alexa سيكون مشروعًا جيداً لعلوم البيانات. أتمنى أن تكون قد أحببت هذه المقالة حول مهمة تحليل آراء Amazon Alexa باستخدام بايثون.

43) نظام توصية Amazon باستخدام بايثون Amazon Recommendation System using Python

تعد أنظمة التوصيات (Recommendation Systems) أحد التطبيقات المستخدمة على نطاق واسع لعلوم البيانات في معظم الشركات استنادًا إلى المنتجات والخدمات عبر الإنترنت. Amazon مثال رائع لهذه الشركات. كونك موقعًا للتسوق عبر الإنترنت، تحتاج Amazon إلى إنشاء توصيات مخصصة لتوفير تجربة مستخدم أفضل. في هذه المقالة، سوف أطلعك على كيفية إنشاء نظام توصية Amazon باستخدام بايثون.

نظام توصية Amazon

يتبع نظام التوصيات في Amazon مبدأ إنشاء توصيات قائمة على المنتج (product based recommendations) مما يعني قياس أوجه التشابه بين منتجين ثم التوصية بالمنتجات الأكثر تشابهًا لكل مستخدم. لطالما كانت طرق قياس أوجه التشابه بين منتجين محل تركيز رئيسي للباحثين.

ولكن عندما يتعلق الأمر بموقع مثل Amazon، فإنه يحتاج إلى إضافة المزيد من المعايير للتوصية بالمنتجات للمستخدمين مثل جودة المنتج. سيكون للمنتج الجيد دائمًا مجموعة جيدة من المراجعات حتى نتمكن من استخدام كل من نقاط التشابه ومراجعات المنتج لإنشاء توصيات. في القسم أدناه، سوف أطلعك على كيفية إنشاء نظام توصيات أمازون باستخدام بايثون.

نظام توصية Amazon باستخدام بايثون

سأحاول استخدام عدد أقل من مكتبات بايثون التي يمكنني استخدامها لإنشاء نظام التوصية هذا. للعمل مع البيانات، سأستخدم مكتبة Pandas و NumPy فقط في بايثون. فلنستورد البيانات ونرى كيفية إنشاء نظام توصيات Amazon باستخدام بايثون:

مجموعة البيانات:

```
import numpy as np
import pandas as pd

data = pd.read_csv("amazon.csv")
print(data.head())
```

```
     AKM1MP6P0OYPR  0132793040  5.0  1365811200
0   A2CX7LUOHB2NDG  0321732944  5.0  1341100800
1   A2NWSAGRHCP8N5  0439886341  1.0  1367193600
2   A2WNBOD3WNDNKT  0439886341  3.0  1374451200
3   A1GI0U4ZRJA8WN  0439886341  1.0  1334707200
4   A1QGNMC6O1VW39  0511189877  5.0  1397433600
```

لا تحتوي مجموعة البيانات التي أستخدمها هنا على أسماء أعمدة، لذا دعنا نعطي الأسماء الأكثر ملاءمة لهذه الأعمدة:

```
data.columns = ['user_id', 'product_id','ratings','timestamp']
```

مجموعة البيانات هذه كبيرة جداً لذا سأختار عينة:

```
df = data[:int(len(data) * .1)]
```

دعنا الآن نجهز مجموعة البيانات لإنشاء نظام توصية:

```
counts = df['user_id'].value_counts()
data = df[df['user_id'].isin(counts[counts >= 50].index)]
data.groupby('product_id')['ratings'].mean().sort_values(ascen
ding=False)
final_ratings = data.pivot(index = 'user_id', columns
='product_id', values = 'ratings').fillna(0)

num_of_ratings = np.count_nonzero(final_ratings)
possible_ratings = final_ratings.shape[0] *
final_ratings.shape[1]
density = (num_of_ratings/possible_ratings)
density *= 100
final_ratings_T = final_ratings.transpose()

grouped = data.groupby('product_id').agg({'user_id':
'count'}).reset_index()
grouped.rename(columns = {'user_id': 'score'},inplace=True)
training_data = grouped.sort_values(['score', 'product_id'],
ascending = [0,1])
training_data['Rank'] =
training_data['score'].rank(ascending=0, method='first')
recommendations = training_data.head()
```

سأكتب الآن دالة بايثون لإنشاء توصيات بناءً على درجة (score) تقييمات المنتج:

```
def recommend(id):
    recommend_products = recommendations
    recommend_products['user_id'] = id
    column = recommend_products.columns.tolist()
    column = column[-1:] + column[:-1]
    recommend_products = recommend_products[column]
    return recommend_products

print(recommend(11))
```

	user_id	product_id	score	Rank
113	11	B00004SB92	6	1.0
1099	11	B00008OE6I	5	2.0
368	11	B00005AW1H	4	3.0
612	11	B0000645C9	4	4.0
976	11	B00007KDVI	4	5.0

الملخص

هذه هي الطريقة التي يمكننا بها إنشاء نظام التوصية من Amazon باستخدام بايثون. لا تحتوي مجموعة البيانات هذه على أسماء منتجات بداخلها، بل تحتوي فقط على معرّف المنتج، لذا تصبح نتيجة مراجعات المنتج أهم ميزة لمثل هذه الأنواع من مجموعات البيانات. أتمنى أن تعجبك هذه المقالة حول كيفية إنشاء نظام توصية Amazon باستخدام بايثون.

المصادر

1. 125 Data Science Projects You Can Try with Python, Aman Kharwal, https://python.plainenglish.io/85-data-science-projects-c03c8750599e.
2. 40+ Data Analysis Projects with Python, Aman Kharwal, https://amankharwal.medium.com/data-analysis-projects-with-python-a262a6f9e68c.

Data Science

By Example

40 Data Science Projects Solved and Explained with Python

By: Dr. Alaa Taima